To appear in *Engineering Optimization*
Vol. 00, No. 00, Month 20XX, 1–26

## Original Article

# A shifted hyperbolic augmented Lagrangian-based artificial fish two swarm algorithm with guaranteed convergence for constrained global optimization

Ana Maria A.C. Rocha[a*], M. Fernanda P. Costa[b] and Edite M.G.P. Fernandes[a]

[a]*Algoritmi Research Centre, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal*;
[b]*Centre of Mathematics, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal*

(*v4.1 released April 2013*)

This article presents a shifted hyperbolic penalty function and proposes an augmented Lagrangian-based algorithm for nonconvex constrained global optimization problems. Convergence to an $\varepsilon$-global minimizer is proved. At each iteration $k$, the algorithm requires the $\varepsilon^{(k)}$-global minimization of a bound constrained optimization subproblem, where $\varepsilon^{(k)} \to \varepsilon$. The subproblems are solved by a stochastic population-based metaheuristic that relies on the artificial fish swarm paradigm and a two swarm strategy. To enhance the speed of convergence, the algorithm invokes the Nelder-Mead local search with a dynamically defined probability. Numerical experiments with benchmark functions and engineering design problems are presented. The results show that the proposed shifted hyperbolic augmented Lagrangian compares favorably with other deterministic and stochastic penalty-based methods.

**Keywords:** global optimization; augmented Lagrangian; shifted hyperbolic penalty; artificial fish swarm; Nelder-Mead search

## 1. Introduction

This article presents a shifted hyperbolic augmented Lagrangian algorithm for solving nonconvex optimization problems subject to inequality constraints. The algorithm aims at guaranteeing that a global optimal solution of the problem is obtained, up to a required accuracy $\varepsilon > 0$. The mathematical formulation of the problem is:

$$\min_{x\in\Omega} f(x) \quad \text{subject to} \quad g(x) \le 0 \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^p$ are nonlinear continuous functions, possibly nondifferentiable, and $\Omega = \{x \in \mathbb{R}^n : -\infty < l \le x \le u < \infty\}$. Functions $f$ and $g$ may be nonconvex and many local minima may exist in the feasible region. For the class of global optimization problems, methods based on penalty functions are common in the literature. In this type of methods, the constraint violation is combined with the objective function to define a penalty function. This function aims at penalizing infeasible solutions by increasing their fitness values proportionally to their level of constraint violation. In Ali, Golalikhani, and Zhuang (2014); Ali and Zhu (2013); Barbosa and Lemonge (2008); Coello (2002); Lemonge, Barbosa, and Bernardino (2015); Mezura-Montes and Coello (2011); Silva, Barbosa, and Lemonge (2011), penalty methods and stochastic approaches are used to generate a population of points, at each iteration, aiming to explore the search space and to converge to a global optimal solution.

*Corresponding author. Email: arocha@dps.uminho.pt

Metaheuristics are approximate methods or heuristics that are designed to search for good solutions, known as near-optimal solutions, with less computational effort and time than the more classical algorithms. While heuristics are tailored to solve a specific problem, metaheuristics are general-purpose algorithms that can be applied to solve almost any optimization problem. They do not take advantage of any specificity of the problem and can then be used as black boxes. They are usually non-deterministic and their behavior do not dependent on problem's properties. Some generate just one solution at a time, *i.e.*, at each iteration, like the simulated annealing, variable neighborhood search and tabu search, others generate a set of solutions at each iteration, improving them along the iterative process. These population-based metaheuristics have been used to solve a variety of optimization problems (Boussaïd, Lepagnot, and Siarry 2013) from combinatorial to the continuous ones. Popular metaheuristics are the differential evolution (DE), electromagnetism-like mechanism (EM), genetic algorithm (GA), harmony search (HS), and the challenging swarm intelligence-based algorithms, such as, ant colony optimization, artificial bee colony, artificial fish swarm, firefly algorithm and particle swarm optimization (PSO) (see Akay and Karaboga (2012); Kamarian et al. (2014); Mahdavi, Fesanghary, and Damangir (2007); Tahk, Woo, and Park (2007); Tsoulos (2009)).

The artificial fish swarm (AFS) algorithm is one of the swarm intelligence algorithms that has been the subject of intense research in the last decade, *e.g.* M. Costa, Rocha, and Fernandes (2014); Rocha, Fernandes, and Martins (2011); Rocha, Martins, and Fernandes (2011); Yazdani et al. (2013). Recently, the AFS algorithm has been hybridized with other metaheuristics, like the PSO (Tsai and Lin 2011; Zhao et al. 2014), and even with the classic Powell local descent algorithm in (Zhang and Luo 2013). It has also been applied to engineering system design (see Lobato and Steffen (2014)), in 0–1 multidimensional knapsack problems (Azad, Rocha, and Fernandes 2014) and in other cases (see Neshat et al. (2014)).

Augmented Lagrangians (Bertsekas 1999) are penalty functions for which a finite penalty parameter value is sufficient to guarantee convergence to the solution of the constrained problem. Methods based on augmented Lagrangians have similarities to penalty methods in that they find the optimal solution of the constrained optimization problem by identifying the optimal solutions of a sequence (or possible just one) of unconstrained subproblems (C. Wang and Li 2009; Zhou and Yang 2012). Recent studies regarding augmented Lagrangians and stochastic methods are available in Ali and Zhu (2013); L. Costa, Espírito Santo, and Fernandes (2012); Deb and Srivastava (2012); Jansen and Perez (2011); Long et al. (2013); Rocha and Fernandes (2011); Rocha, Martins, and Fernandes (2011). Recently, a hyperbolic augmented Lagrangian paradigm has been presented in M. Costa, Rocha, and Fernandes (2014). The therein augmented Lagrangian function is different from the one herein proposed and the subproblems are approximately solved by a standard AFS algorithm.

In the present study, the properties of a shifted hyperbolic penalty are derived and discussed. The convergence properties of an augmented Lagrangian algorithm proving that every accumulation point of a sequence of iterates generated by the shifted hyperbolic augmented Lagrangian algorithm is feasible and is an $\varepsilon$-global minimizer of problem (1), where $\varepsilon$ is a sufficiently small positive value, are analyzed. The developed algorithm uses a new AFS algorithm that generates two subpopulations of points and move them differently aiming to explore the search space and to avoid local optimal solutions. To enhance convergence, an intensification phase based on the Nelder-Mead local search procedure is invoked with a dynamically defined probability.

The article is organized as follows. In Section 2, the shifted hyperbolic penalty function is presented and the augmented Lagrangian algorithm and its convergence properties are derived. Section 3 describes the AFS algorithm and discusses the new algorithm that incorporates the two swarm paradigm and the heuristic to invoke the Nelder-Mead local search. Finally, Section 4 presents some numerical experiments and the article is concluded in Section 5.

## 2. Shifted hyperbolic penalty-based augmented Lagrangian algorithm

Here, the good properties of the 2-parameter hyperbolic penalty term (see Xavier (2001)) given by

$$\mathbf{P}_h(g_i(x);\tau,\rho) = \tau g_i(x) + \sqrt{\tau^2\,(g_i(x))^2 + \rho^2} \tag{2}$$

are extended, where $g_i(x)$ is the $i$th constraint function of the problem (1) ($i = 1,\ldots,p$) and $\tau \geq 0$ and $\rho \geq 0$ are penalty parameters, to a shifted penalty term. The penalty in (2), which is a continuously differentiable function with respect to $x$, is made to work in Xavier (2001) as follows. In the initial phase of the process, $\tau$ increases while $\rho$ remains constant, causing a significant increase of the penalty at infeasible points. This way the search is directed to the feasible region since the goal is to minimize the penalty. From the moment that a feasible point is obtained, the penalty parameter $\rho$ decreases, while $\tau$ remains constant.

The herein derived methodology uses a shifted penalty strategy in which one is willing to modify the origin from which infeasibility is to be penalized, *i.e.*, one penalizes the positive deviation of $g_i(x)$ with respect to a certain threshold value, $T_i$, rather than 0. If $T_i = -\dfrac{\delta_i}{\tau}$ is defined then using (2) the shifted hyperbolic penalty term arises

$$\mathbf{P}_s(g_i(x);\delta_i,\tau,\mu) = \tau\left(g_i(x) + \frac{\delta_i}{\tau} + \sqrt{\left(g_i(x) + \frac{\delta_i}{\tau}\right)^2 + \mu^2}\right) \tag{3}$$

where $\delta_i$, the $i$th component of the vector $\delta = (\delta_1,\ldots,\delta_p)^T$, is the multiplier associated with the inequality constraint $g_i(x) \leq 0$, and $\tau$ and $\mu \in \mathbb{R}$ are the penalty parameters, being $\mu = (\rho/\tau) \geq 0$, for $\tau \neq 0$. For any $i$, the function $\mathbf{P}_s(g_i(x);\delta_i,\tau,\mu)$ approaches the straight line $l_1(g_i(x)) = 0$ as $g_i(x) \to -\infty$ (the horizontal asymptote), and to the straight line $l_2(g_i(x)) = 2\tau g_i(x) + 2\delta_i$ as $g_i(x) \to +\infty$ (the oblique asymptote), for $\tau > 0, \delta_i > 0$ and $\mu > 0$, as shown in Figure 1.

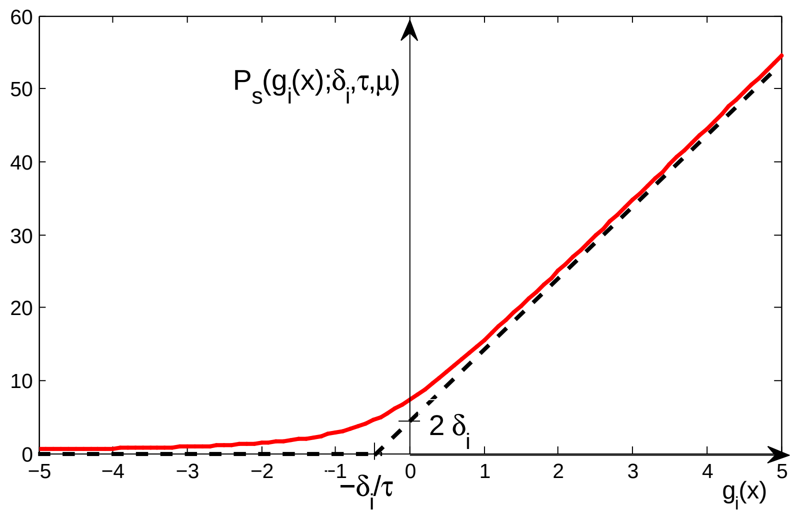


Figure 1. Shifted hyperbolic penalty function

The most important properties that will be used in the present study are now listed.

**Properties.** Let $y \equiv g_i(x)$ and $\delta_y \equiv \delta_i$ (fixed) for any $i$. Function in (3) satisfies the following properties:

P1: $\mathbf{P}_s(y;\delta_y,\tau,\mu)$ is continuously differentiable with respect to $y \in \mathrm{R}$, for $\tau,\mu > 0$ and $\delta_y \geq 0$;
P2: for $y_1, y_2 \in \mathrm{R}$, $y_1 < y_2 \leq 0$, $0 \leq \mathbf{P}_s(y_1;\delta_y,\tau,\mu) < \mathbf{P}_s(y_2;\delta_y,\tau,\mu)$, for fixed $\tau > 0$ and $\delta_y,\mu \geq 0$;
P3: for $y_1, y_2 \in \mathrm{R}$, $0 < y_1 < y_2$, $0 < \mathbf{P}_s(y_1;\delta_y,\tau,\mu) < \mathbf{P}_s(y_2;\delta_y,\tau,\mu)$, for fixed $\tau > 0$ and $\delta_y,\mu \geq 0$;
P4: $2\delta_y \leq \mathbf{P}_s(0;\delta_y,\tau,\mu) = \delta_y + \sqrt{\delta_y^2 + (\tau\mu)^2} \leq 2\delta_y + \tau\mu$, for fixed $\tau > 0$ and $\delta_y,\mu \geq 0$;
P5: $$\mathbf{P}_s(y;\delta_y,\tau,0) = \begin{cases} 0, & \text{if } y < 0 \text{ and } y + \frac{\delta_y}{\tau} \leq 0 \\ 2\tau\left(y + \frac{\delta_y}{\tau}\right), & \text{if } y < 0 \text{ and } y + \frac{\delta_y}{\tau} > 0 \\ 2\tau\left(y + \frac{\delta_y}{\tau}\right), & \text{if } y > 0 \end{cases}$$
for fixed $\tau > 0$ and $\delta_y \geq 0$;
P6: $$\mathbf{P}_s(y;\delta_y,\tau,\mu) \leq \begin{cases} \tau\mu, & \text{if } y < 0 \text{ and } y + \frac{\delta_y}{\tau} \leq 0 \\ 2\delta_y + \tau\mu, & \text{if } y < 0 \text{ and } y + \frac{\delta_y}{\tau} > 0 \\ 2\tau\left(y + \frac{\delta_y}{\tau}\right) + \tau\mu, & \text{if } y > 0 \end{cases}$$
for fixed $\tau > 0$ and $\delta_y,\mu \geq 0$.

The penalty term (3) is to be used in an augmented Lagrangian based method (Bertsekas 1999) where the augmented Lagrangian function, hereafter denoted by shifted hyperbolic augmented Lagrangian, has the form

$$\phi(x;\delta,\tau,\mu) = f(x) + \sum_{i=1}^{p} \mathbf{P}_s(g_i(x);\delta_i,\tau,\mu). \tag{4}$$

The implemented multiplier method penalizes the inequality constraints of the problem (1) while the bound constraints are always satisfied when solving the subproblems. This means that, at each iteration $k$, where $k$ denotes the iteration counter of the outer cycle, for fixed values of the multipliers vector $\delta^{(k)}$, and penalty parameters $\tau^{(k)}$ and $\mu^{(k)}$, the subproblem is the bound constrained optimization problem:

$$\min_{x\in\Omega} \phi(x;\delta^{(k)},\tau^{(k)},\mu^{(k)}) \equiv f(x) + \sum_{i=1}^{p} \tau^{(k)} \left( g_i(x) + \frac{\delta_i^{(k)}}{\tau^{(k)}} + \sqrt{\left(g_i(x) + \frac{\delta_i^{(k)}}{\tau^{(k)}}\right)^2 + (\mu^{(k)})^2} \right). \tag{5}$$

When penalty terms (2) and (3) are added to the objective function, they aim to assign a high cost to infeasible points. As the penalty parameter $\tau^{(k)}$ increases, problem (5) approximates the original constrained problem. While the use of the penalty term (2) corresponds to taking the multipliers to be equal to zero, and the success of the penalty algorithm depends only on increasing the penalty parameter to infinity (with the decreasing of the parameter $\mu^{(k)}$), when (3) is used the performance of the algorithm may be improved by using nonzero multiplier approximates $\delta^{(k)}$ that are updated after solving the subproblem (5). Hence, when (3) is used, the algorithm is able to compute optimal multiplier values and provides information about which constraints are active at the optimal solution and the relative importance of each constraint to the optimal solution.

When the optimization problem is nonconvex, a global optimization method is required to solve the subproblem (5), so that the algorithm has some guarantee to converge to a global solution instead of being trapped in a local one. The following is remarked:

*Remark 1* When finding the global minimum of a continuous objective function $\phi(x)$ over a bounded space $\Omega \subset \mathrm{R}^n$, the point $\bar{x} \in \Omega$ is a solution to the minimization problem if $\phi(\bar{x}) \leq \min_{y\in\Omega} \phi(y) + \varepsilon$, where $\varepsilon$ is the error bound which reflects the accuracy required for the solution.

Thus, at each iteration $k$ of the outer cycle, an $\varepsilon^{(k)}$-global solution of subproblem (5) is required. Some challenging differences between augmented Lagrangian-based algorithms are located on the algorithm used to find the sequence of approximate solutions to the subproblems (5). The proposal for computing an $\varepsilon^{(k)}$-global minimum of subproblem (5), for fixed values of $\delta^{(k)}, \tau^{(k)}, \mu^{(k)}$ is based on a stochastic population-based algorithm, denoted by enhanced artificial fish two swarm (AF-2S) algorithm. Other important differences are related with the updating of penalty, smooth or tolerance parameters aiming to promote the algorithm's convergence.

### 2.1 *Augmented Lagrangian-based algorithm*

A formal description of the shifted hyperbolic augmented Lagrangian (shifted-HAL) algorithm for solving the original problem (1), is presented in Algorithm 1.

**Algorithm 1** Shifted-HAL algorithm

**Require:** $\tau^{(1)} \geq 1, \mu^{(1)} > 0, \gamma_\tau > 1, 0 < \gamma_\mu < 5\gamma_\mu \leq (\gamma_\tau)^{-1}, 0 < \gamma_\varepsilon, \gamma_\eta < 1, 0 < \varepsilon \ll 1, \varepsilon^{(1)} > \varepsilon,$ $\eta^{(1)} > 0, LB, \delta_{\max} \in [0, +\infty), \delta_i^{(1)} \in [0, \delta_{\max}]$ for all $i = 1, \dots, p$.

Set $k = 1$
Randomly generate $m$ points in $\Omega$
**repeat**
  Find an $\varepsilon^{(k)}$-global minimizer $x^{(k)}$ of subproblem (5) such that:

$$\phi(x^{(k)}; \delta^{(k)}, \tau^{(k)}, \mu^{(k)}) \leq \phi(x; \delta^{(k)}, \tau^{(k)}, \mu^{(k)}) + \varepsilon^{(k)} \text{ for all } x \in \Omega. \tag{6}$$

  Compute $\delta_i^{(k+1)}$ using (9), for $i = 1, \dots, p$
  **if** $\|V^{(k)}\| \leq \eta^{(k)}$ **then**
    Set $\tau^{(k+1)} = \tau^{(k)}$, $\mu^{(k+1)} = \gamma_\mu \mu^{(k)}$, $\varepsilon^{(k+1)} = \max\left\{\varepsilon, \gamma_\varepsilon \varepsilon^{(k)}\right\}$, $\eta^{(k+1)} = \gamma_\eta \eta^{(k)}$
  **else**
    Set $\tau^{(k+1)} = \gamma_\tau \tau^{(k)}$, $\mu^{(k+1)} = 5\gamma_\mu \mu^{(k)}$, $\varepsilon^{(k+1)} = \varepsilon^{(k)}$, $\eta^{(k+1)} = \gamma_\eta \eta^{(1)}$
  **end if**
  Set $k = k + 1$
**until** $\|V^{(k)}\| = 0$ and $f(x^{(k)}) \leq LB + \varepsilon$

The form of the shifted penalty in (3) compels that the penalty parameter $\mu$ is to be decreased at all iterations, whatever the proximity to the feasible region, although a strong reduction is beneficial when feasibility is approaching. Thus, the strategy is the following. When the level of feasibility and complementarity at the iterate $x^{(k)}$ is acceptable within the tolerance $\eta^{(k)} > 0$,

$$\|V^{(k)}\| \leq \eta^{(k)}, \tag{7}$$

where

$$V_i^{(k)} = \max\left\{g_i(x^{(k)}), -\frac{\delta_i^{(k+1)}}{\tau^{(k)}}\right\}, i = 1, \dots, p, \tag{8}$$

the parameter $\tau^{(k)}$ remains unchanged since there is no need to penalize further the violation, but $\mu^{(k+1)} = \gamma_\mu \mu^{(k)}$, for $0 < \gamma_\mu < 1$, *i.e.*, $\mu^{(k)}$ undergoes a significant change, referred to as a fast reduction, to accelerate the movement towards feasibility. Otherwise, $\tau$ is increased using $\tau^{(k+1)} = \gamma_\tau \tau^{(k)}$ where $\gamma_\tau > 1$, and a slow reduction is imposed on $\mu$: $\mu^{(k+1)} = 5\gamma_\mu \mu^{(k)}$. If $\gamma_\mu < 5\gamma_\mu \leq (\gamma_\tau)^{-1}$ is chosen in the algorithm, $\{\tau^{(k)}\mu^{(k)}\}$ is a bounded monotonically decreasing sequence that converges to zero (see Gonzaga and Castillo (2003)).

Note that when the penalty parameter is not updated, the tolerances for solution quality and feasibility, $\varepsilon$ and $\eta$ respectively, are decreased as follows: $\varepsilon^{(k+1)} = \max\left\{\varepsilon, \gamma_\varepsilon \varepsilon^{(k)}\right\}$ and $\eta^{(k+1)} = \gamma_\eta \eta^{(k)}$, where $0 < \gamma_\varepsilon, \gamma_\eta < 1$, so that an even better solution than its ancestor is computed. It is required that $\{\eta^{(k)}\}$ defines a decreasing sequence of positive values converging to zero as $k \to \infty$. On the other hand, when penalty $\tau$ is increased aiming to further penalize constraints violation, $\varepsilon$ remains unchanged ($\varepsilon^{(k+1)} = \varepsilon^{(k)}$) and $\eta$ is allowed even to increase (relative to its previous value), $\eta^{(k+1)} = \gamma_\eta \eta^{(1)}$, where $\eta^{(1)} > 0$ is the user provided initial value. In this study, $\|\cdot\|$ represents the Euclidean norm.

The multipliers vector is estimated using the usual first-order updating scheme (Bertsekas 1999) coupled with a safeguarded procedure:

$$\delta_i^{(k+1)} = \max\left\{0, \min\left\{\delta_i^{(k)} + \tau^{(k)} g_i(x^{(k)}), \delta_{\max}\right\}\right\}, \tag{9}$$

for $i = 1, \ldots, p$ where $\delta_{\max} \in [0, +\infty)$. The use of this safeguarding procedure by projecting the multipliers onto a suitable bounded interval aims to ensure boundedness of the sequence. The algorithm terminates when a solution $x^{(k)}$ that is feasible, satisfies the complementarity condition and has an objective function value within $\varepsilon$ of the known minimum is found, *i.e.*, when $\|V(x^{(k)})\| = 0$ and $f(x^{(k)}) \leq LB + \varepsilon$ for a sufficiently small tolerance $\varepsilon > 0$, where $LB$ denotes the smallest function value considering all algorithms that found a feasible solution of problem (1).

## 2.2 *Convergence to an $\varepsilon$-global minimizer*

Here, it is proved that every accumulation point, denoted by $x^*$, of the sequence $\{x^{(k)}\}$, produced by the shifted-HAL algorithm is an $\varepsilon$-global minimizer of problem (1). Since the set $\Omega$ is compact and the augmented Lagrangian function $\phi(x; \delta^{(k)}, \tau^{(k)}, \mu^{(k)})$ is continuous, the $\varepsilon^{(k)}$-global minimizer of subproblem (5), $x^{(k)}$, does exist. For this convergence analysis, the methodology presented in Birgin, Floudas, and Martínez (2010) is followed, where differentiability and convexity properties are not required. The assumptions that are needed to show convergence of the shifted-HAL algorithm (Algorithm 1) to an $\varepsilon$-global minimum are now stated. Some of them are common in the convergence analysis of augmented Lagrangian methods for constrained global optimization, *e.g.* (Birgin, Floudas, and Martínez 2010).

As previously mentioned, the augmented Lagrangian approach is combined with a stochastic population-based algorithm for solving the subproblems (5), and therefore it is assumed that the sequence $\{x^{(k)}\}$ is well defined (see Assumption A 2.2 below). Note that Assumption A 2.4 is concerned with $\delta^*$, the Lagrange multiplier vector at $x^*$ and Assumption A 2.5 below is a consequence of the way the two parameters $\tau^{(k)}$ and $\mu^{(k)}$ are updated in the algorithm.

Assumption A 2.1 A global minimizer $z$ of the problem (1) exists.

Assumption A 2.2 The sequence $\{x^{(k)}\}$ generated by the Algorithm 1 is well defined and there exists a subset of indices $\mathcal{N} \subseteq \mathrm{N}$ so that $\lim_{k \in \mathcal{N}} x^{(k)} = x^*$.

Assumption A 2.3 The functions $f : \mathrm{R}^n \to \mathrm{R}$ and $g : \mathrm{R}^n \to \mathrm{R}^p$ are continuous on $\Omega$.

Assumption A 2.4 For all $i = 1, \ldots, p$, there exists $\delta_{\max} \in [0, +\infty)$ such that $\delta_i^* \in [0, \delta_{\max}]$.

Assumption A 2.5 $\{\tau^{(k)} \mu^{(k)}\}$ is a bounded and monotonically decreasing sequence of non-negative real numbers.

First, it is proved that every accumulation point of the sequence $\{x^{(k)}\}$ is feasible.

THEOREM 2.6 *Assume that Assumptions A 2.1 through A 2.3 and A 2.5 hold. Then every accumulation point $x^*$ of the sequence $\{x^{(k)}\}$ produced by the Algorithm 1 is feasible for problem* (1).

*Proof* Since $x^{(k)} \in \Omega$ and $\Omega$ is closed then $x^* \in \Omega$. Two cases are considered: (a) $\{\tau^{(k)}\}$ is bounded; (b) $\tau^{(k)} \to \infty$.

In case (a), there exists an index $K$ and a value $\bar{\tau} > 0$ such that $\tau^{(k)} = \bar{\tau}$ for all $k \geq K$. This means that, for all $k \geq K$, condition $\|V^{(k)}\| \leq \eta^{(k)}$ is satisfied. Since $\eta^{(k)} \to 0$, according to (8), either $g_i(x^{(k)}) \to 0$ or $\delta_i^{(k+1)} \to 0$ with $g_i(x^{(k)}) \leq 0$, for all $i = 1, \ldots, p$. Thus, by Assumptions A 2.2 and A 2.3, $g_i(x^*) \leq 0$ for all $i$, and the accumulation point is feasible.

The proof in case (b) is made by contradiction by assuming that $x^*$ is not feasible and using $z$ (see Assumption A 2.1), the same for all $k$, such that $g_i(z) \leq 0$ for all $i$. Let $I^+$, $I_1^-$, $I_2^-$ and $I_3^-$ be index subsets of $I = \{1, \ldots, p\}$ defined by:

- $I^+ = \{i : g_i(x^*) > 0 \geq g_i(z)\}$,
- $I_1^- = \{i : g_i(z) \leq g_i(x^*) \leq 0\}$,
- $I_2^- = \{i : g_i(x^*) \leq g_i(z) \leq 0 \text{ and } g_i(z) + \frac{\delta_i^{(k)}}{\tau^{(k)}} > 0\}$,
- $I_3^- = \{i : g_i(x^*) \leq g_i(z) \leq 0 \text{ and } g_i(z) + \frac{\delta_i^{(k)}}{\tau^{(k)}} \leq 0\}$

such that $I = I^+ \cup I_1^- \cup I_2^- \cup I_3^-$ and $I^+ \neq \emptyset$ (since $x^*$ is infeasible). Using the properties P2 and P3 described above, then

$$\sum_{i \in I^+ \cup I_1^-} \left( g_i(x^*) + \frac{\delta_i^{(k)}}{\tau^{(k)}} + \sqrt{\left( g_i(x^*) + \frac{\delta_i^{(k)}}{\tau^{(k)}} \right)^2 + \left(\mu^{(k)}\right)^2} \right) \geq \sum_{i \in I^+ \cup I_1^-} \left( g_i(z) + \frac{\delta_i^{(k)}}{\tau^{(k)}} + \sqrt{\left( g_i(z) + \frac{\delta_i^{(k)}}{\tau^{(k)}} \right)^2 + \left(\mu^{(k)}\right)^2} \right)$$

and similarly

$$\sum_{i \in I_2^- \cup I_3^-} \left( g_i(x^*) + \frac{\delta_i^{(k)}}{\tau^{(k)}} + \sqrt{\left( g_i(x^*) + \frac{\delta_i^{(k)}}{\tau^{(k)}} \right)^2 + \left(\mu^{(k)}\right)^2} \right) \leq \sum_{i \in I_2^- \cup I_3^-} \left( g_i(z) + \frac{\delta_i^{(k)}}{\tau^{(k)}} + \sqrt{\left( g_i(z) + \frac{\delta_i^{(k)}}{\tau^{(k)}} \right)^2 + \left(\mu^{(k)}\right)^2} \right)$$

hold, since $\delta^{(k)}$ and $\mu^{(k)}$ are bounded by definition (see (9) and updates in Algorithm 1). Using Assumptions A 2.2 and A 2.3, for a large enough $k \in \mathcal{N}$, there exists a positive constant $c$ such that

$$\tau^{(k)} \sum_{i \in I^+ \cup I_1^-} \left( g_i(x^{(k)}) + \frac{\delta_i^{(k)}}{\tau^{(k)}} + \sqrt{\left( g_i(x^{(k)}) + \frac{\delta_i^{(k)}}{\tau^{(k)}} \right)^2 + \left(\mu^{(k)}\right)^2} \right) \geq \tau^{(k)} \sum_{i \in I^+ \cup I_1^-} \left( g_i(z) + \frac{\delta_i^{(k)}}{\tau^{(k)}} + \sqrt{\left( g_i(z) + \frac{\delta_i^{(k)}}{\tau^{(k)}} \right)^2 + \left(\mu^{(k)}\right)^2} \right) + \tau^{(k)} c$$

and therefore

$$\begin{aligned}\sum_{i=1}^{p}\mathbf{P}_s(g_i(x^{(k)});\delta_i^{(k)},\tau^{(k)},\mu^{(k)}) \geq &\sum_{i=1}^{p}\mathbf{P}_s(g_i(z);\delta_i^{(k)},\tau^{(k)},\mu^{(k)}) + \sum_{i\in I_2^-\cup I_3^-}\mathbf{P}_s(g_i(x^{(k)});\delta_i^{(k)},\tau^{(k)},\mu^{(k)})\\ &-\sum_{i\in I_2^-}\mathbf{P}_s(g_i(z);\delta_i^{(k)},\tau^{(k)},\mu^{(k)}) - \sum_{i\in I_3^-}\mathbf{P}_s(g_i(z);\delta_i^{(k)},\tau^{(k)},\mu^{(k)})\\ &+\tau^{(k)}c.\end{aligned}$$

Using the properties P2 and P6,

$$\begin{aligned}\sum_{i=1}^{p}\mathbf{P}_s(g_i(x^{(k)});\delta_i^{(k)},\tau^{(k)},\mu^{(k)}) \geq &\sum_{i=1}^{p}\mathbf{P}_s(g_i(z);\delta_i^{(k)},\tau^{(k)},\mu^{(k)})\\ &-\sum_{i\in I_2^-}\left(2\delta_i^{(k)}+\tau^{(k)}\mu^{(k)}\right) - \sum_{i\in I_3^-}\tau^{(k)}\mu^{(k)}+\tau^{(k)}c\\ \geq &\sum_{i=1}^{p}\mathbf{P}_s(g_i(z);\delta_i^{(k)},\tau^{(k)},\mu^{(k)}) - 2\sum_{i\in I_2^-}\delta_i^{(k)} - p\tau^{(k)}\mu^{(k)}+\tau^{(k)}c\end{aligned}$$

and

$$\begin{aligned}f(x^{(k)})+\sum_{i=1}^{p}\mathbf{P}_s(g_i(x^{(k)});\delta_i^{(k)},\tau^{(k)},\mu^{(k)}) \geq\ & f(z)+\sum_{i=1}^{p}\mathbf{P}_s(g_i(z);\delta_i^{(k)},\tau^{(k)},\mu^{(k)}) - 2\sum_{i\in I_2^-}\delta_i^{(k)}\\ &-p\tau^{(k)}\mu^{(k)}+\tau^{(k)}c+f(x^{(k)})-f(z)\end{aligned}$$

are obtained. Using Assumptions A 2.3 and A 2.5, for large enough $k\in\mathcal{N}$ $(\tau^{(k)}\to\infty)$,

$$-2\sum_{i\in I_2^-}\delta_i^{(k)} - p\tau^{(k)}\mu^{(k)}+\tau^{(k)}c+f(x^{(k)})-f(z) > \varepsilon^{(k)} > 0,$$

implying that $\phi(x^{(k)};\delta^{(k)},\tau^{(k)},\mu^{(k)}) > \phi(z;\delta^{(k)},\tau^{(k)},\mu^{(k)})+\varepsilon^{(k)}$, which contradicts the definition of $x^{(k)}$ in (6). ■

Now, it is proved that a sequence of iterates generated by the algorithm converges to an $\varepsilon$-global minimizer of problem (1).

THEOREM 2.7 *Assume that the Assumptions A 2.1 through A 2.5 hold. Then every accumulation point $x^*$ of a sequence $\{x^{(k)}\}$ generated by Algorithm 1 is an $\varepsilon$-global minimizer of problem* (1).

*Proof* Two cases are considered: (a) $\{\tau^{(k)}\}$ is bounded; (b) $\tau^{(k)}\to\infty$.

First, the case (a). By the definition of $x^{(k)}$ in the Algorithm 1, and since $\tau^{(k)}=\bar{\tau}$ for all $k\geq K$, one gets

$$\begin{aligned}&f(x^{(k)})+\bar{\tau}\sum_{i=1}^{p}\left(g_i(x^{(k)})+\frac{\delta_i^{(k)}}{\bar{\tau}}+\sqrt{\left(g_i(x^{(k)})+\frac{\delta_i^{(k)}}{\bar{\tau}}\right)^2+\left(\mu^{(k)}\right)^2}\right)\\ &\qquad\leq f(z)+\bar{\tau}\sum_{i=1}^{p}\left(g_i(z)+\frac{\delta_i^{(k)}}{\bar{\tau}}+\sqrt{\left(g_i(z)+\frac{\delta_i^{(k)}}{\bar{\tau}}\right)^2+\left(\mu^{(k)}\right)^2}\right)+\varepsilon^{(k)}\end{aligned} \tag{10}$$

where $z \in \Omega$ comes from Assumption A 2.1. Since $g_i(z) \leq 0$ and $\delta_i^{(k)} \geq 0$ for all $i$, and $\mu^{(k)}, \bar{\tau} > 0$, using the properties P4 and P6, then for all $k \geq K$

$$\begin{aligned} f(x^{(k)}) + \sum_{i=1}^{p} \mathbf{P}_s(g_i(x^{(k)}); \delta_i^{(k)}, \bar{\tau}, \mu^{(k)}) &\leq f(z) + \sum_{i \in I^0} \left(2\delta_i^{(k)} + \bar{\tau}\mu^{(k)}\right) + \sum_{i \in I_1^-} \left(2\delta_i^{(k)} + \bar{\tau}\mu^{(k)}\right) \\ &\quad + \sum_{i \in I_2^-} \bar{\tau}\mu^{(k)} + \varepsilon^{(k)} \\ &\leq f(z) + 2\sum_{i=1}^{p} \delta_i^{(k)} + \sum_{i=1}^{p} \bar{\tau}\mu^{(k)} + \varepsilon^{(k)} \end{aligned}$$

holds, where $I^0$, $I_1^-$ and $I_2^-$ are subsets of $I$ defined by:

- $I^0 = \{i : g_i(z) = 0\}$,
- $I_1^- = \{i : g_i(z) < 0 \text{ and } g_i(z) + \frac{\delta_i^{(k)}}{\bar{\tau}} > 0\}$,
- $I_2^- = \{i : g_i(z) < 0 \text{ and } g_i(z) + \frac{\delta_i^{(k)}}{\bar{\tau}} \leq 0\}$

and $I^- = I_1^- \cup I_2^-$. Now, let $\mathcal{N}_1 \subset \mathcal{N}$ be a subset of indices such that $\lim_{k \in \mathcal{N}_1} \delta^{(k)} = \delta^*$. Taking limits for $k \in \mathcal{N}_1$ and using $\lim_{k \in \mathcal{N}_1} \varepsilon^{(k)} = \varepsilon$ and Assumption A 2.2,

$$f(x^*) + \sum_{i=1}^{p} \mathbf{P}_s(g_i(x^*); \delta_i^*, \bar{\tau}, \mu^{(k)}) \leq f(z) + 2\sum_{i=1}^{p} \delta_i^* + p\bar{\tau}\mu^{(k)} + \varepsilon$$

is obtained. Since $x^*$ is feasible and $\delta_i^* \geq 0$ (finite by Assumption A 2.4) for all $i$, using

- for $i \in I^0 \subseteq I$ such that $g_i(x^*) = 0$ (property P4 above):

$$2\sum_{i \in I^0} \delta_i^* \leq \sum_{i \in I^0} \mathbf{P}_s(g_i(x^*); \delta_i^*, \bar{\tau}, \mu^{(k)});$$

- for $i \in I^- \subseteq I$ such that $g_i(x^*) < 0$:

$$0 \leq \sum_{i \in I^-} \mathbf{P}_s(g_i(x^*); \delta_i^*, \bar{\tau}, \mu^{(k)});$$

one gets

$$f(x^*) + 2\sum_{i \in I^0} \delta_i^* \leq f(z) + 2\sum_{i=1}^{p} \delta_i^* + p\,\bar{\tau}\mu^{(k)} + \varepsilon.$$

Therefore

$$f(x^*) \leq f(z) + 2\sum_{i \in I^-} \delta_i^* + p\,\bar{\tau}\mu^{(k)} + \varepsilon.$$

Since condition (7) and Algorithm 1 require that $\lim_{k \to \infty} V_i^{(k)} = 0$ then $\delta_i^* = 0$ when $g_i(x^*) < 0$ (see (8)). Finally, using $\lim_{k \in \mathcal{N}_1} \mu^{(k)} = 0$ (recall property P5) then $f(x^*) \leq f(z) + \varepsilon$ which proves the claim that $x^*$ is an $\varepsilon$-global minimizer, since $z$ is a global minimizer.

For case (b), $\phi(x^{(k)}; \delta^{(k)}, \tau^{(k)}, \mu^{(k)}) \leq \phi(z; \delta^{(k)}, \tau^{(k)}, \mu^{(k)}) + \varepsilon^{(k)}$ for all $k \in \mathrm{N}$. Since $z$ is feasible, recalling (10) (with $\bar{\tau}$ replaced by $\tau^{(k)}$) and using property P6, one gets

$$f(x^{(k)}) + \sum_{i=1}^{p} \mathbf{P}_s(g_i(x^{(k)}); \delta_i^{(k)}, \tau^{(k)}, \mu^{(k)}) \le f(z) + 2\sum_{i=1}^{p} \delta_i^{(k)} + p\,\tau^{(k)}\mu^{(k)} + \varepsilon^{(k)}.$$

Now, taking limits for $k \in \mathscr{N}$, using Assumptions A 2.2, A 2.3, A 2.4 and A 2.5, $\lim_{k\in\mathscr{N}} \varepsilon^{(k)} = \varepsilon$ and the same reasoning as above, one gets

$$f(x^*) \le f(z) + 2\sum_{i=1}^{p} \delta_i^* - 2\sum_{i\in I^0} \delta_i^* + \varepsilon,$$

and the desired result $f(x^*) \le f(z) + 2\sum_{i\in I^-} \delta_i^* + \varepsilon = f(z) + \varepsilon$ is obtained. ■

## 3. Enhancing the AF-2S algorithm with Nelder-Mead local search

For solving the subproblems (5), a new AFS algorithm, denoted by enhanced AF-2S algorithm is proposed. AFS is a stochastic population-based algorithm for global optimization (see Rocha, Fernandes, and Martins (2011); Rocha, Martins, and Fernandes (2011)). The enhanced AF-2S algorithm combines the global search AFS method with a two swarm paradigm and an intensification phase based on the Nelder-Mead local search (Nelder and Mead 1965) procedure. The goal here is to find an approximate global minimizer $x^{(k)}$ of the subproblem (5) satisfying (6).

For simplicity, $\phi^k(x)$ is used to denote the objective function of the subproblem (5), instead of $\phi(x; \delta^{(k)}, \tau^{(k)}, \mu^{(k)})$. The position of a point in the space is represented by $x_j \in \mathrm{R}^n$ (the $j$th point of a population) and $m < \infty$ is the number of points in the population. Let $x_{\text{best}}$ be the best point of a population of $m$ points so that (for fixed $\delta^{(k)}, \tau^{(k)}$ and $\mu^{(k)}$):

$$\phi_{\text{best}}^k \equiv \phi^k(x_{\text{best}}) = \min\{\phi^k(x_j), j = 1, \ldots, m\} \tag{11}$$

is the corresponding function value and $x_j$ $(j = 1, \ldots, m)$ are the points of the population.

### 3.1 *Standard AFS algorithm*

The AFS algorithm is now briefly described. At each iteration $t$, the current population of $m$ points, herein denoted by $x_1, x_2, \ldots, x_m$ is used to generate a trial population $y_1, y_2, \ldots, y_m$. Initially, the population is randomly generated in the entire search space $\Omega$. Each fish/point $x_j$ movement is defined according to the number of points inside its 'visual scope'. The 'visual scope' is the closed neighborhood centered at $x_j$ with a positive radius $\upsilon$ which varies with the point progress. A fraction of the maximum distance between $x_j$ and the other points $x_l$, $l \neq j$, $\upsilon_j = \max_l \|x_j - x_l\|$ is used.

When the 'visual scope' is empty, a Random Behavior is performed, in which the trial $y_j$ is selected along a random direction starting from $x_j$. When the 'visual scope' is crowded, and a point randomly selected from the visual, $x_{\text{rand}}$, has a better fitness, $\phi^k(x_{\text{rand}}) < \phi^k(x_j)$, the Searching Behavior is implemented, *i.e.*, $y_j$ is randomly generated along the direction from $x_j$ to $x_{\text{rand}}$. Otherwise, the Random Behavior is performed. When the 'visual scope' is not crowded, and the best point inside the 'visual scope', $x_{\min}$, has a better fitness than $x_j$, the Chasing Behavior is performed. This means that $y_j$ is randomly generated along the direction from $x_j$ to $x_{\min}$. However, if $x_{\min}$ is not better than $x_j$, the Swarming Behavior may be tried instead. Here, the central point of the 'visual scope', $\bar{x}$, is computed and if it has better fitness than $x_j$, $y_j$ is computed randomly along the direction from $x_j$ to $\bar{x}$; otherwise, a point $x_{\text{rand}}$ is randomly selected from the 'visual scope' and if it has a better fitness than $x_j$ the Searching Behavior is implemented. However, if $\phi^k(x_{\text{rand}}) \geq \phi^k(x_j)$ a Random Behavior is performed. Note that in either case, each point $x_j$ will produce a trial point denoted by $y_j$.

Finally, to choose which point between the current $x_j$ and the trial $y_j$ will be a point of the population for the next iteration, a Selection Procedure is carried out. The current point is replaced by its trial if $\phi^k(y_j) \leq \phi^k(x_j)$; otherwise the current point is preserved.

### 3.2 *Artificial fish two swarm algorithm*

In order to improve the capability of searching the space for promising regions where the global minimizers lie, this study presents a new fish swarm-based proposal that defines two subpopulations (or swarms), hereafter denoted by artificial fish two swarm – with acronym AF-2S. Each swarm moves differently, but they may share information with each other: one is the 'master swarm' and the other is the 'training swarm'. The 'master swarm' aims to explore the search space more effectively, defining trial points from the current ones using random numbers drawn from a stable but heavy-tailed distribution, thus providing occasionally long movements. Depending on the number of points inside the 'visual scope' of each point $x_j$ of the 'training swarm', the corresponding trial point is produced by the standard artificial fish behaviors.

To be able to produce a trial $y_j$, from the current $x_j$, ideas like those of Bare-bones particle swarm optimization in Kennedy and Eberhart (2001) and the model for mutation in evolutionary programming (Lee and Yao 2004) may be used:

$$(y_j)_i = \gamma + \sigma \mathbf{Y}_i \tag{12}$$

where $\gamma$ represents the center of the distribution that may be given by $(x_j)_i$ or $((x_j)_i + (x_{\text{best}})_i)/2$, $\sigma$ represents the distance between $(x_j)_i$ and $(x_{\text{best}})_i$, and each $\mathbf{Y}_i$ is an identically distributed random variable from the Gaussian distribution with mean 0 and variance 1. Note that $\mathbf{Y}$ may be the random variable of another probability distribution. Here, the standard Lévy distribution is proposed since it can search a wider area of the search space and generate more distinct values in the search space than the Gaussian distribution. This way the likelihood of escaping from local optima is higher. The Lévy distribution, denoted by $\mathbf{L}_i(\alpha, \beta, \gamma, \sigma)$, is characterized by four parameters. The parameter $\beta$ gives the skewness ($\beta = 0$ means that the shape is symmetric relative to the mean). The shape of the Lévy distribution can be controlled with $\alpha$. For $\alpha = 2$ it is equivalent to the Gaussian distribution, whereas for $\alpha = 1$ it is equivalent to the Cauchy distribution. The distribution is stable for $\alpha = 0.5$ and $\beta = 1$. $\sigma$ is the scale parameter and is used to describe the variation relative to the center of the distribution. The location parameter $\gamma$ gives the center. When $\gamma = 0$ and $\sigma = 1$, the standard form, simply denoted by $\mathbf{L}(\alpha)$ when $\beta = 0$, is obtained.

Hence, the proposal for further exploring the search space and improve efficiency is the following. The points from the 'master swarm' always move according to the Lévy distribution, *i.e.*, each trial point $y_j$ is generated component by component ($i = 1, \ldots, n$) as follows:

$$(y_j)_i = \begin{cases} (x_j)_i + (\sigma_j)_i \mathbf{L}_i(\alpha) & \text{if } rand() \leq p \\ (x_{\text{best}})_i + (\sigma_j)_i \mathbf{L}_i(\alpha) & \text{otherwise} \end{cases} \tag{13}$$

where $(\sigma_j)_i = \left|(x_j)_i - (x_{\text{best}})_i\right|/2$ and $x_{\text{best}}$ is the best point of the population, whatever the swarm it belongs to. $\mathbf{L}_i(\alpha)$ denotes a number that is generated following the standard Lévy distribution with the parameter $\alpha = 0.5$, for each $i$, $rand()$ is a random number generated uniformly from $[0, 1]$ and $p$ is a user specified probability value for sampling around the best point to occur. On the other hand, each point in the 'training swarm' moves according to the classical AFS behaviors. Initially, the points of the 'master swarm' are randomly selected from the population and remain in the same swarm over the course of optimization. The same is true for the points in the 'training swarm'.

### 3.3 *Enhanced AF-2S algorithm*

The enhanced AF-2S algorithm herein presented integrates a deterministic direct search method, known as Nelder-Mead (N-M) algorithm, *e.g.* (McKinnon 1998; Nelder and Mead 1965), into the AF-2S algorithm. Further, since a two-swarm-based strategy is implemented, cooperation and competition between the two swarms is required in order to reduce the computational effort.

The issues to address when a local search is integrated into a population-based algorithm are: (i) which local search method should be chosen; (ii) which points of the population should be used in the local search; (iii) how frequently should the local search be invoked; (iv) how many function evaluations should be computed. To address issue (i), the N-M algorithm, originally published in 1965 (Nelder and Mead 1965), is chosen due to its popularity in multidimensional derivative-free optimization. Regarding issues (ii) and (iii), the most common strategy invokes the local search algorithm at every iteration and applies it only to the best point of the population. Since the N-M local search requires $n+1$ points to define the vertices of the simplex, the strategy collects the best point of the population (both swarms together), $x_{\text{best}}$, and the $n$ best points from the swarm that does not include $x_{\text{best}}$, hereafter designated by $z_i, i = 1,\ldots,n$. Since the 'master swarm' is assumed to be the smallest of the two swarms, the number of points, $m_M$, should be at least $n$, so that $n$ points may be supplied to the N-M local search, when $x_{\text{best}}$ belongs to the 'training swarm'. Thus, the cooperation feature of the enhanced AF-2S algorithm is present when the N-M local search is invoked since both swarms contribute to the points required by N-M, to define the vertices of the initial simplex. Furthermore, competition arises when the best point produced by the N-M algorithm, $x_{\text{best}}^{N-M}$, replaces the worst point of the population, $x_{\text{worst}}$. Note that the best points in both swarms – $x_{\text{best}}$ in one swarm and the $n$ best points in the other – are able to enhance the worst point of the population, either it belongs to the first swarm or to the second. The previously designated $z_i, i = 1,\ldots,n$ are updated by the remaining $n$ points $x_i^{N-M}, i = 1,\ldots,n$ generated by the N-M. Algorithm 2 describes the pseudo-code for the enhanced AF-2S algorithm, from which the N-M local search may be invoked, with a certain probability.

To define the frequency of invoking the N-M local search, the algorithm relies on a dynamically defined probability

$$p_{N-M} = \begin{cases} \dfrac{1}{1+\sum_{i=1}^{n} \phi^k(z_i)/n - \phi_{\text{best}}^k} & \text{if } \phi_{\text{best}}^k < \xi \\ \dfrac{\phi_{\text{best}}^k}{\sum_{i=1}^{n} \phi^k(z_i)/n} & \text{otherwise} \end{cases}, \tag{14}$$

for $\xi = 0.00001$. The proposed probability-based heuristic aims to avoid the local intensification phase when $p_{N-M}$ approaches one, and to follow the local exploitation when $p_{N-M} \approx 0$. Note that the smallest the $p_{N-M}$ the further away are the points $z_i$ (on average) from the best point, thus requiring some improvement. The goal of this methodology is to reduce the overall computational effort without perturbing the speed of convergence of the algorithm. Invoking the N-M local search procedure is a task that requires between one to $n$ extra function evaluations per N-M iteration (see McKinnon (1998)). In this study, the N-M algorithm terminates when a pre-specified number of function evaluations, $NF_{\max}$, is achieved.

**Algorithm 2** Enhanced AF-2S algorithm

**Require:** $m$, $x_j, j = 1, \ldots, m$, $m_M$, $t_{\max}$, $LB$, $\delta^{(k)}$, $\tau^{(k)}$, $\mu^{(k)}$, $\varepsilon^{(k)}$ and $NF_{\max}$
Set $t = 0$ ; Select $x_{\text{best}}$ according to (11)
Randomly select $m_M$ points of the population to define the 'master swarm' ; Define $\mathscr{M}$ with the set of indices of the points in the 'master swarm'
**while** $\phi^k_{\text{best}} \geq LB + \varepsilon^{(k)}$ and $t \leq t_{\max}$ **do**
  **for** $j = 1, \ldots, m$ **do**
    **if** $j \in \mathscr{M}$ **then**
      Compute $y_j$ by Lévy distribution using (13) with $p = 0.5$
    **else if** 'visual scope' is empty **then**
      Compute $y_j$ by Random Behavior
    **else if** 'visual scope' is crowded **then**
      **if** $\phi^k(x_{\text{rand}}) < \phi^k(x_j)$ **then**
        Compute $y_j$ by Searching Behavior
      **else**
        Compute $y_j$ by Random Behavior
      **end if**
    **else if** $\phi^k(x_{\min}) < \phi^k(x_j)$ **then**
      Compute $y_j$ by Chasing Behavior
    **else if** $\phi^k(\bar{x}) < \phi^k(x_j)$ **then**
      Compute $y_j$ by Swarming Behavior
    **else if** $\phi^k(x_{\text{rand}}) < \phi^k(x_j)$ **then**
      Compute $y_j$ by Searching Behavior
    **else**
      Compute $y_j$ by Random Behavior
    **end if**
  **end for**
  **for** $j = 1, \ldots, m$ **do**
    **if** $\phi^k(y_j) \leq \phi^k(x_j)$ **then**
      Set $x_j = y_j$
    **end if**
  **end for**
  Select $x_{\text{best}}$ and the worst point $x_{\text{worst}}$ ; Set $t = t + 1$
  Select $z_i, i = 1, \ldots, n$ for the local search ; Compute $p_{N-M}$ according to (14)
  **if** $rand() < 1 - p_{N-M}$ **then**
    Run Nelder-Mead algorithm until $NF_{\max}$ is reached ; Set $x_{\text{worst}} = x^{N-M}_{\text{best}}$
    **for** $i = 1, \ldots, n$ **do**
      Set $z_i = x^{N-M}_i$
    **end for**
  **end if**
**end while**

## 4. Numerical experiments

For a practical validation of the shifted-HAL algorithm based on the enhanced AF-2S algorithm for solving the subproblems (5), three sets of benchmark problems are selected. First, a set of 20 small constrained global optimization problems are tested, where the number of variables ranges from 2 to 6 and the number of constraints ranges from 1 to 12 (see Birgin, Floudas, and Martínez (2010)). Second, six problems that have inequality constraints only (C01, C07, C08, C13, C14 and C15) from the suit of scalable functions developed for the 'CEC 2010 Competition on Constrained Real-Parameter Optimization' are selected from (Mallipeddi and Suganthan 2010a), and finally six well-known engineering design problems are used to analyze the performance of the algorithm when integer and continuous variables are present. The C programming language is used in this real-coded algorithm and the computational tests were performed on a PC with a 2.7 GHz Core i7-4600U and 8 Gb of memory. Although the shifted hyperbolic penalty function is meant to work with inequality constrained optimization problems, it has been possible to handle problems with equality constraints, $h(x) = 0$, as long as they are reformulated into the following couple of inequality constraints $h(x) - \upsilon \leq 0$ and $-h(x) - \upsilon \leq 0$, where $\upsilon > 0$ is the tolerance used for the constraints relaxation. The parameters have been set after a set of testing experiments: $\upsilon = 10^{-5}$, $\tau^{(1)} = 10$, $\gamma_\tau = \sqrt{10}$, $\mu^{(1)} = 1$, $\gamma_\mu = 0.05$, $\varepsilon^{(1)} = \eta^{(1)} = 1$, $\gamma_\varepsilon = \gamma_\eta = 0.1$, $\delta_i^{(1)} = 1$ for all $i = 1,\ldots,p$ and $\delta_{\max} = 10^4$. The probability $p$ in the definition (13) is set to 0.5, the maximum number of function evaluations in the N-M algorithm is $NF_{\max} = 100$. Since the parameter update schemes available in the outer cycle have a moderate influence on the convergence speed of the shifted-HAL algorithm, $k_{\max} = 50$ and $t_{\max} = 12$ are considered. To stop the Algorithm 1, an error tolerance of $10^{-6}$ is used in the feasibility condition and $\varepsilon$ is set to $10^{-5}$. If one of the conditions in the stopping rule is not satisfied, the algorithm will run for a maximum of $k_{\max}$ iterations. Unless otherwise stated, $m = 5n$ is set and each problem is solved 30 times. Setting $m_M = \lfloor \frac{m}{2} \rfloor$ turned out to be a good choice. A large 'master swarm' has empowered the exploratory ability of the AF-2S algorithm, improving the consistency of the solutions and reducing the overall number of function evaluations.

The performance of the shifted-HAL algorithm, based on the Algorithm 2 with the N-M local search, is compared with another hyperbolic augmented Lagrangian algorithm (HAL-AFS) presented in M. Costa, Rocha, and Fernandes (2014) as well as with two deterministic methods, namely the exact penalty DIRECT-based method (e.penalty-DIRECT) in Di Pillo, Lucidi, and Rinaldi (2012) and the augmented Lagrangian $\alpha$BB-based method (AL-$\alpha$BB) of Birgin, Floudas, and Martínez (2010).

Table 1 lists the number of the problem as shown in Birgin, Floudas, and Martínez (2010), 'Prob'; the best solution obtained by the tested augmented Lagrangian algorithm during the 30 runs, '$f_{\text{best}}$'; the median (as a measure of the central tendency of the distribution) of the 30 solutions, '$f_{\text{med}}$'; the number of function evaluations, 'n.f.e.', and the CPU time (in seconds), 'time', required to reach the reported $f_{\text{best}}$. The table also displays the value of the constraint violation at the best solution, 'C.V.$_{\text{best}}$' and the average number of iterations in the outer cycle (over the 30 runs), '$It_{\text{av}}$'. The solution found by e.penalty-DIRECT, '$f^*$', a measure of the constraint violation, 'C.V.' (in Di Pillo, Lucidi, and Rinaldi (2012)), the number of outer iterations, '$It$', required by AL-$\alpha$BB and the solution reported in Birgin, Floudas, and Martínez (2010) (used as the best-known solution available in the literature), '$LB$', are also shown. Using suitable reformulations (Birgin, Floudas, and Martínez 2010; Di Pillo, Lucidi, and Rinaldi 2012) data related with $n$ (number of variables) and $nc$ (number of equality and inequality constraints) are also displayed.

From the results it is concluded that the proposed shifted-HAL algorithm based on the Algorithm 2, with the N-M local search, performs reasonably well. Not all the solutions obtained for problems 2(b) and 3(a), during the 30 runs, are as good as it is expected, when compared with those shown in Birgin, Floudas, and Martínez (2010), but some are better than the ones reported in Di Pillo, Lucidi, and Rinaldi (2012). For the other problems, the results of the current study are very competitive. It can be concluded that the algorithm is quite consistent with similar values for '$f_{\text{best}}$' and '$f_{\text{med}}$' for most tested problems. It is noteworthy that the results in terms of number of function evaluations and CPU time are very competitive, taking into account that this is a population-based method rather than a pointwise strategy. To analyze the statistical significance of the results the Wilcoxon signed-rank test is used. This is a non-parametric statistical test for testing hypothesis on median. The Matlab$^{\text{TM}}$ (Matlab is a registered trademark of the MathWorks, Inc.) function `signrank` is used. This function returns a logical value: '1' indicates a rejection of the null hypothesis that the data are observations from a distribution with a certain median, '0' indicates a failure to reject the null hypothesis, at a 0.05 significance level. The comparisons are made between the median solution computed from the results and (i) the median solution when comparing with HAL-AFS, (ii) the value of $f^*$ when comparing with e.penalty-DIRECT, (iii) and the *LB* when comparing with AL-$\alpha$BB. The character '$*$' in the table indicates a rejection of the null hypothesis with a p-value $< 0.05$, *i.e.*, that the result is statistically different.

Based on the results of the Table 1, it can be concluded that the shifted augmented Lagrangian algorithm (that relies on the enhanced AF-2S for solving the subproblems (5)) will also compete favorably with the penalty-based stochastic algorithms presented in Ali, Golalikhani, and Zhuang (2014); Deb and Srivastava (2012); Silva, Barbosa, and Lemonge (2011). This conclusion is based on the fact that the proposed algorithm outperforms the HAL-AFS algorithm presented in M. Costa, Rocha, and Fernandes (2014) which in turn performed well when compared with those stochastic algorithms.

The second set of problems comprises six problems that arise from well-known scalable test functions with rotated constraints and were used in the "2010 Congress on Evolutionary Computation Competition on Constrained Real-Parameter Optimization". Twenty five independent runs are performed and the allowed maximum number of function evaluations (n.f.$\text{e}_{\text{max}}$) is 2E+05. The results are compared with those of the winner of the competition (Takahama and Sakai 2010) where an $\varepsilon$ constrained differential evolution combined with an archive and gradient-based mutation,'$\varepsilon$DEg', is used, as well as with another algorithm, 'ECHT-DE', a differential evolution with ensemble of constraint handling techniques that ranked second between twelve competitors (see Mallipeddi and Suganthan (2010b)). In Table 2, $f_{\text{best}}$, $f_{\text{med}}$, $f_{\text{av}}$ (the average of the 25 obtained solutions) and the standard deviation, 'St.d($f$)', are shown for each problem when 'n.f.e.' reaches 2E+04, 1E+05 and 2E+05 (along three consecutive rows of the table). Furthermore, the number of violated constraints, 'n.v.c.', at the median solution by more than 1E+00, by more than 1E-02 and by more than 1E-04 have been reported. The mean value of the constraint violations at the median solution, '$v_{\text{av}}$', has been saved. The Wilcoxon signed-rank test is used again to analyze the statistical significance relative to the median values. The comparisons are made between the median solution computed from the current results and each of the median solutions of both articles (Takahama and Sakai 2010) and (Mallipeddi and Suganthan 2010b). The character '$*$' indicates that the result is statistically different with a p-value $< 0.05$. Note that all the results produced by the algorithm and reported in the table are feasible and therefore n.v.c. = (0,0,0) and $v_{\text{av}}$ = 0.0E+00. This comparison corroborates the competitive performance of the proposed algorithm when solving constrained global optimization problems.

Finally, the next experiment aims to show the effectiveness of the proposed algorithm when solving more complex and real application engineering design problems with discrete, integer and continuous variables. Engineering problems with mixed-integer design variables are quite common. To handle integer variables, a simple heuristic that relies on the rounding off to the nearest integer before evaluation and selection stages is implemented. For the discrete variables, the used heuristic rounds to the nearest value of the discrete set during the evaluation and selection stages. Six well-known engineering design problems are considered. For the first four problems, all parameter settings are the same as the previous experiment that produced the results of Table 1, except that $t_{\max} = 25$.

**Welded Beam Design Problem**

The design of a welded beam (Hedar and Fukushima 2006; Lee and Geem 2005; Silva, Barbosa, and Lemonge 2011) is the most used problem to assess the effectiveness of an algorithm. The objective is to minimize the cost of a welded beam, subject to constraints on the shear stress, bending stress in the beam, buckling load on the bar, end deflection of the beam, and side constraints. There are four continuous design variables and five inequality constraints.

Table 1. Numerical results for comparison

| | | HAL-AFS | | | | shifted-HAL with enhanced AF-2S | | | | | | e.penalty-DIRECT | | | | AL-$\alpha$BB | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Prob | $(n,nc)$ | $f_{\text{best}}$ | n.f.e. | time | $f_{\text{med}}$ | $f_{\text{best}}$ | n.f.e. | time | C.V.$_{\text{best}}$ | $f_{\text{med}}$ | $It_{\text{av}}$ | $f^*$ | n.f.e. | time | C.V. | time | $It$ | $LB$ |
| 1 | (5,3) | 0.0342 | 9608 | 0.046 | 0.1204∗ | 0.0294 | 9723 | 0.024 | 2.58E-06 | 0.0450 | 50 | 0.0625∗ | 39575 | 0.328 | 2.4E-07 | 18.86 | 9 | 0.029313∗ |
| 2(a) | (5,10) | -380.674 | 15813 | 0.109 | -369.111∗ | -400.0000 | 2434 | 0.005 | 0.0E+00 | -400.0000 | 21 | -134.1127∗ | 115107 | 2.078 | 8.4E-04 | 0.13 | 8 | -400.00 |
| 2(b) | (5,10) | -385.051 | 15808 | 0.093 | -360.786∗ | -600.0000 | 4038 | 0.008 | 0.0E+00 | -400.0000 | 42 | -768.4569∗ | 120057 | 3.828 | 5.3E-04 | 0.76 | 13 | -600.00∗ |
| 2(c) | (5,10) | -743.416 | 15612 | 0.109 | -693.743∗ | -750.0000 | 2433 | 0.004 | 0.0E+00 | -750.0000 | 13 | -82.9774∗ | 102015 | 0.953 | 8.4E-04 | 0.16 | 8 | -750.00 |
| 2(d) | (5,12) | -399.910 | 15394 | 0.094 | -399.492∗ | -400.0000 | 2711 | 0.007 | 0.0E+00 | -400.0000 | 12 | -385.1704∗ | 229773 | 2.328 | 0.0E+00 | 0.23 | 4 | -400.00 |
| 3(a) | (6,5) | -0.3880 | 18928 | 0.109 | -0.3849∗ | -0.3840 | 19144 | 0.040 | 3.6E-06 | -0.3691 | 50 | -0.3861∗ | 48647 | 1.234 | 1.0E-06 | 12.07 | 6 | -0.38880∗ |
| 3(b) | (2,1) | -0.3888 | 2589 | 0.000 | -0.3888∗ | -0.3888 | 548 | 0.001 | 0.0E+00 | -0.3888 | 15 | -0.3888∗ | 3449 | 0.031 | 0.0E+00 | 2.90 | 4 | -0.38881∗ |
| 4 | (2,1) | -6.6667 | 2242 | 0.000 | -6.6667 | -6.6667 | 698 | 0.001 | 0.0E+00 | -6.6667 | 12 | -6.6666∗ | 3547 | 0.031 | 0.0E+00 | 0.00 | 4 | -6.6666∗ |
| 5 | (3,3) | 201.159 | 2926 | 0.000 | 201.159 | 201.1593 | 2717 | 0.003 | 6.5E-07 | 201.1593 | 15 | 201.1593 | 14087 | 0.078 | 1.7E-04 | 0.04 | 7 | 201.16∗ |
| 6 | (2,1) | 376.292 | 5617 | 0.000 | 376.293∗ | 376.2919 | 1578 | 0.003 | 0.0E+00 | 376.2919 | 13 | 0.4701∗ | 1523 | 0.000 | 2.1E-05 | 0.01 | 5 | 376.29∗ |
| 7 | (2,4) | -2.8284 | 3434 | 0.000 | -2.8284 | -2.8284 | 886 | 0.001 | 0.0E+00 | -2.8284 | 10 | -2.8058∗ | 13187 | 0.125 | 0.0E+00 | 0.02 | 4 | -2.8284∗ |
| 8 | (2,2) | -118.705 | 2884 | 0.000 | -118.705∗ | -118.7049 | 995 | 0.001 | 0.0E+00 | -118.7049 | 11 | -118.7044∗ | 7621 | 0.046 | 0.0E+00 | 0.15 | 6 | -118.70∗ |
| 9 | (6,6) | -13.4018 | 5732 | 0.031 | -13.4017∗ | -13.4019 | 1437 | 0.003 | 0.0E+00 | -13.4019 | 6 | -13.4026∗ | 68177 | 2.171 | 1.4E-04 | 0.00 | 1 | -13.402∗ |
| 10 | (2,2) | 0.7418 | 6342 | 0.015 | 0.7418∗ | 0.7418 | 1155 | 0.001 | 0.0E+00 | 0.7418 | 15 | 0.7420∗ | 6739 | 0.078 | 0.0E+00 | 0.01 | 4 | 0.74178 |
| 11 | (2,1) | -0.5000 | 3313 | 0.015 | -0.5000 | -0.5000 | 1043 | 0.002 | 0.0E+00 | -0.5000 | 21 | -0.5000 | 3579 | 0.031 | 0.0E+00 | 0.01 | 4 | -0.50000 |
| 12 | (2,1) | -16.7389 | 98 | 0.000 | -16.7389 | -16.7389 | 267 | 0.001 | 0.0E+00 | -16.7389 | 5 | -16.7389 | 3499 | 0.015 | 5.4E-06 | 0.01 | 8 | -16.739∗ |
| 13 | (3,2) | 189.345 | 9230 | 0.031 | 189.347∗ | 189.3449 | 9703 | 0.012 | 9.0E-06 | 189.3449 | 50 | 195.9553∗ | 8085 | 0.078 | 9.2E-04 | 0.47 | 8 | 189.35∗ |
| 14 | (4,3) | -4.5142 | 6344 | 0.031 | -4.5142∗ | -4.5142 | 1170 | 0.002 | 0.0E+00 | -4.5142 | 7 | -4.3460∗ | 19685 | 0.250 | 9.2E-05 | 0.00 | 1 | -4.5142 |
| 15 | (3,3) | 0.0000 | 2546 | 0.015 | 0.0000 | 0.0000 | 3187 | 0.004 | 1.0E-05 | 0.0000 | 20 | 0.0000 | 1645 | 0.000 | 4.9E-05 | 0.06 | 4 | 0.0000 |
| 16 | (5,3) | 0.7049 | 1850 | 0.015 | 0.7049∗ | 0.7049 | 371 | 0.001 | 0.0E+00 | 0.7049 | 6 | 0.7181∗ | 22593 | 0.312 | 2.0E-04 | 0.15 | 6 | 0.70492 |

Table 2. Results for comparison with Takahama and Sakai (2010) and Mallipeddi and Suganthan (2010b)

| Prob | $n.f.e_{max}$ | shifted-HAL with enhanced AF-2S | | | | results† in Takahama and Sakai (2010) | | | | results† in Mallipeddi and Suganthan (2010b) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $f_{best}$ | $f_{med}$ | $f_{av}$ | St.d.$(f)$ | $f_{best}$ | $f_{med}$ | $f_{av}$ | St.d.$(f)$ | $f_{best}$ | $f_{med}$ | $f_{av}$ | St.d.$(f)$ |
| C01 | 2E+04 | -6.892E-01 | -6.254E-01 | -6.125E-01 | 5.4E-02 | -7.471E-01 | -7.466E-01∗ | -7.462E-01 | 1.7E-03 | -6.462E-01 | -5.392E-01∗ | -5.478E-01 | 4.3E-02 |
| | 1E+05 | -7.402E-01 | -6.862E-01 | -6.824E-01 | 3.5E-02 | -7.473E-01 | -7.473E-01∗ | -7.470E-01 | 1.3E-03 | -7.473E-01 | -7.473E-01∗ | -7.470E-01 | 1.4E-03 |
| | 2E+05 | -7.404E-01 | -6.912E-01 | -6.959E-01 | 3.7E-02 | -7.473E-01 | -7.473E-01∗ | -7.470E-01 | 1.3E-03 | -7.473E-01 | -7.473E-01∗ | -7.470E-01 | 1.4E-03 |
| C07 | 2E+04 | 2.955E+00 | 7.139E+01 | 1.206E+02 | 1.6E+02 | 4.804E+00 | 6.574E+00∗ | 6.973E+00 | 1.7E+00 | 5.195E+00 | 7.591E+00 | 9.403E+00 | 1.1E+01 |
| | 1E+05 | 9.400E-11 | 4.139E-02 | 2.103E+01 | 4.5E+01 | 4.381E-18 | 8.660E-17∗ | 1.323E-16 | 1.6E-16 | 0.000E+00 | 0.000E+00∗ | 1.329E-01 | 7.3E-01 |
| | 2E+05 | 0.000E+00 | 1.870E-08 | 4.198E-01 | 1.0E+00 | 0.000E+00 | 0.000E+00∗ | 0.000E+00 | 0.0E+00 | 0.000E+00 | 0.000E+00∗ | 1.329E-01 | 7.3E-01 |
| C08 | 2E+04 | 1.013E+00 | 8.993E+01 | 1.445E+02 | 1.8E+02 | 1.165E+01 | 3.365E+01 | 3.940E+01 | 2.8E+01 | 7.492E+01 | 2.518E+02∗ | 3.619E+02 | 2.8E+02 |
| | 1E+05 | 1.810E-09 | 4.176E+00 | 2.865E+01 | 4.7E+01 | 1.469E-18 | 1.094E+01 | 6.728E+00 | 5.6E+00 | 0.000E+00 | 7.098E+00 | 6.157E+00 | 6.5E+00 |
| | 2E+05 | 0.000E+00 | 4.795E-06 | 4.543E+00 | 1.6E+01 | 0.000E+00 | 1.094E+01∗ | 6.728E+00 | 5.6E+00 | 0.000E+00 | 7.098E+00∗ | 6.157E+00 | 6.5E+00 |
| C13 | 2E+04 | -6.558E+01 | -5.959E+01 | -5.840E+01 | 5.1E+00 | -2.744E+01 | 8.018E-01∗ | -6.355E+00 | 1.2E+01 | -6.837E+01 | -6.206E+01 | -6.171E+01 | 4.2E+00 |
| | 1E+05 | -6.558E+01 | -5.959E+01 | -5.938E+01 | 5.0E+00 | -6.843E+01 | -6.792E+01∗ | -6.748E+01 | 1.1E+00 | -6.843E+01 | -6.352E+01∗ | -6.512E+01 | 2.4E+00 |
| | 2E+05 | -6.843E+01 | -5.959E+01 | -5.951E+01 | 5.1E+00 | -6.843E+01 | -6.843E+01∗ | -6.843E+01 | 1.0E-06 | -6.843E+01 | -6.352E+01∗ | -6.512E+01 | 2.4E+00 |
| C14 | 2E+04 | 1.170E+00 | 3.736E+02 | 6.198E+03 | 2.1E+04 | 2.929E+01 | 1.839E+02 | 2.209E+02 | 2.0E+02 | 3.207E+07 | 6.214E+10∗ | 1.136E+11 | 2.2E+11 |
| | 1E+05 | 4.300E-11 | 2.772E+02 | 3.889E+03 | 1.8E+04 | 9.522E-16 | 1.937E-14∗ | 6.145E-14 | 1.3E-13 | 0.000E+00 | 0.000E+00∗ | 7.024E+05 | 3.2E+06 |
| | 2E+05 | 0.000E+00 | 2.772E+02 | 3.870E+03 | 1.8E+04 | 0.000E+00 | 0.000E+00∗ | 0.000E+00 | 0.0E+00 | 0.000E+00 | 0.000E+00∗ | 7.024E+05 | 3.2E+06 |
| C15 | 2E+04 | 2.231E-01 | 7.060E+02 | 6.463E+05 | 3.2E+06 | 9.865E+01 | 6.273E+02 | 9.352E+02 | 1.2E+03 | 1.983E+12 | 4.818E+13∗ | 6.852E+13 | 7.1E+13 |
| | 1E+05 | 3.000E-12 | 4.058E+02 | 2.583E+05 | 1.3E+06 | 6.145E-15 | 5.664E-14∗ | 1.799E-01 | 8.8E-01 | 1.456E+07 | 6.740E+10∗ | 2.515E+13 | 5.7E+13 |
| | 2E+05 | 0.000E+00 | 4.058E+02 | 2.580E+05 | 1.3E+06 | 0.000E+00 | 0.000E+00∗ | 1.799E-01 | 8.8E-01 | 0.000E+00 | 1.222E+10∗ | 2.339E+13 | 5.3E+13 |

† The results herein reported from Takahama and Sakai (2010) and Mallipeddi and Suganthan (2010b) have n.v.c. = (0,0,0) and $v_{av}$ = 0.0E+00.

The optimization problem is expressed as follows:

$$\begin{aligned}
&\min\ 1.10471x_1^2x_2 + 0.04811x_3x_4(14+x_2)\\
&\text{subject to } \left((\tau')^2 + \frac{\tau'\tau''x_2}{R} + (\tau'')^2\right)^{1/2} - \tau_{\max} \le 0\,,\ \ \frac{4PL^3}{Ex_4x_3^3} - \delta_{\max} \le 0\,,\ \ x_1 - x_4 \le 0\,,\\
&P - \frac{4.013\left(\frac{EGx_3^2x_4^6}{36}\right)^{1/2}}{L^2}\left(1 - \frac{x_3}{2L}\left(\frac{E}{4G}\right)^{1/2}\right) \le 0\,,\ \ \frac{6PL}{x_4x_3^2} - \sigma_{\max} \le 0\,,
\end{aligned}$$

where $0.125 \le x_1 \le 10$, $0.1 \le x_i \le 10, i = 2,3,4$ and $P = 6000$ lb., $L = 14$ in., $\delta_{\max} = 0.25$ in., $E = 30\times10^6$ psi., $G = 12\times10^6$ psi., $\tau_{\max} = 13600$ psi., $\sigma_{\max} = 30000$[1] psi. and

$$\tau' = \frac{P}{\sqrt{2}x_1x_2},\ \tau'' = \frac{MR}{J},\ M = P\left(L + \frac{x_2}{2}\right),\ R = \left(\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2\right)^{1/2},$$

$$J = \frac{2x_1x_2}{\sqrt{2}}\left(\frac{x_2^2}{12} + \left(\frac{x_1+x_3}{2}\right)^2\right).$$

A comparison of the results with those of the adaptive penalty scheme used within a steady-state genetic algorithm (APS-GA) (Lemonge, Barbosa, and Bernardino 2015), the dynamic use of differential evolution variants combined with the adaptive penalty method (DUVDE+APM) (Silva, Barbosa, and Lemonge 2011), the filter simulated annealing algorithm (FSA), available in Hedar and Fukushima (2006), the hybrid evolutionary algorithm with an adaptive constraint handling technique (HEA-ACT) (Y. Wang et al. 2009), the harmony search metaheuristic (HSm) algorithm (Lee and Geem 2005) and the hybrid version of the electromagnetism-like algorithm (Hybrid EM) (Rocha and Fernandes 2009) is carried out. Table 3 shows the values of the variables and of the objective function of the best run, the average objective function value, $f_{\text{av}}$, and the average number of function evaluations, n.f.e.$_{\text{av}}$, obtained by the shifted-HAL with enhanced AF-2S algorithm and the other methods. The results from APS-GA, DUVDE+APM, FSA, HEA-ACT, HSm and Hybrid EM are taken from the cited articles. It can be seen that the results are very competitive at a reduced computational cost.

Table 3. Comparative results for the welded beam design problem

| Method | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $f_{\text{best}}$ | $f_{\text{av}}$ | n.f.e.$_{\text{av}}$ |
|---|---|---|---|---|---|---|---|
| Shifted-HAL | 0.2444 | 6.2175 | 8.2915 | 0.2444 | 2.380957 | 2.386778 | 25888 |
| APS-GA | 0.2444 | 6.2186 | 8.2915 | 0.2444 | 2.3811 | 2.405 | 50000 |
| DUVDE+APM | 0.2444 | 6.2186 | 8.2915 | 0.2444 | 2.38113 | 2.38113 | 40000 |
| FSA | 0.2444 | 6.2158 | 8.2939 | 0.2444 | 2.381065 | 2.404166 | 56243 |
| HEA-ACT | 0.2444 | 6.2175 | 8.2915 | 0.2444 | 2.380957 | 2.380971 | 30000 |
| HSm | 0.2442 | 6.2231 | 8.2915 | 0.2443 | 2.38 | n.a. | 110000$^\dagger$ |
| Hybrid EM | 0.2435 | 6.1673 | 8.3772 | 0.2439 | 2.386269 | n.a. | 28650$^\dagger$ |

$^\dagger$ number of function evaluations of the best run; n.a. means not available.

## Pressure Vessel Design Problem

The design of a cylindrical pressure vessel with both ends capped with a hemispherical head is to minimize the total cost of fabrication, *e.g.* (Hedar and Fukushima 2006; Silva, Barbosa, and Lemonge 2011). The problem has four design variables and four inequality constraints.

[1]In Lee and Geem (2005), the formulation uses $\sigma_{\max} = 30600$

This is a mixed variables problem where the variables $x_1$, the shell thickness, and $x_2$, the head thickness, are discrete – integer multiples of 0.0625 in. – and the other two are continuous. The mathematical formulation is the following:

$$\begin{aligned} \min\ & 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\ \text{subject to}\ & -x_1 + 0.0193x_3 \leq 0\,,\ \ -x_2 + 0.00954x_3 \leq 0\,, \\ & -\pi x_3^2x_4 - \tfrac{4}{3}\pi x_3^3 + 1296000 \leq 0\,,\ \ x_4 - 240 \leq 0\,, \end{aligned}$$

where $0.1 \leq x_1 \leq 99, 0.1 \leq x_2 \leq 99, 10 \leq x_i \leq 200$, $i = 3,4$. The results of the comparison of the Shifted-HAL algorithm with the APS-GA, DUVDE+APM, FSA, HSm, Hybrid EM, the cost-effective particle swarm optimization (CPSO), presented in Tomassetti (2010), a fish swarm optimization algorithm (FSOA) proposed in Lobato and Steffen (2014), a modified augmented Lagrangian DE-based algorithm (MAL-DE) by Long et al. (2013) and the modified constrained differential evolution (mCDE) in Azad and Fernandes (2013), are available in Table 4. It has been noticed that the results produced by the algorithm are very competitive and require a reasonable amount of function evaluations.

Table 4. Comparative results for the pressure vessel design problem

| Method | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $f_{\text{best}}$ | $f_{\text{av}}$ | n.f.e.$_{\text{av}}$ |
|---|---|---|---|---|---|---|---|
| Shifted-HAL | 0.8125 | 0.4375 | 42.0984 | 176.6366 | 6059.714 | 6059.900 | 26255 |
| APS-GA | 0.8125 | 0.4375 | 42.0984 | 176.6366 | 6059.714 | 6146.822 | 15000 |
| CPSO | 0.8125 | 0.4375 | 42.0984 | 176.6366 | 6059.714 | 6086.9 | *10000*‡ |
| DUVDE+APM | 0.8125 | 0.4375 | 42.0984 | 176.6368 | 6059.718 | 6059.718 | 80000 |
| FSA | 0.7683 | 0.3798 | 39.8096 | 207.2256 | 5868.765 | 6164.586 | 108883 |
| FSOA | 0.8125 | 0.4375 | 42.0913 | 176.7466 | 6061.078 | 6064.726 | 2550 |
| HSm | 1.125 | 0.625 | 58.2789 | 43.7549 | 7198.433 | n.a. | n.a. |
| Hybrid EM | 0.8125 | 0.4375 | 42.0701 | 177.3762 | 6072.232 | n.a. | 20993† |
| MAL-DE | 0.8125 | 0.4375 | 42.0984 | 176.6366 | 6059.714 | 6059.714 | 120000 |
| mCDE | 0.8125 | 0.4375 | 42.0984 | 176.6366 | 6059.714 | n.a. | *1000*‡ |

† number of function evaluations of best run; ‡ number of iterations; n.a. means not available.

Table 5. Comparative results for the speed reducer design problem

| Method | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $f_{\text{best}}$ | $f_{\text{av}}$ | n.f.e.$_{\text{av}}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| Shifted-HAL | 3.5 | 0.7 | 17 | 7.3000 | 7.7153 | 3.3502 | 5.2867 | 2994.355 | 2994.355 | 47113 |
| APS-GA | 3.5 | 0.7 | 17 | 7.3000 | 7.800 | 3.3502 | 5.2867 | 2996.348 | 3007.860 | 36000 |
| CPSO | 3.5 | 0.7 | 17 | 7.3 | 7.8 | 3.3502 | 5.2867 | 2996.348 | 2996.5 | *10000*‡ |
| HEA-ACT | 3.5 | 0.7 | 17 | 7.3004 | 7.7154 | 3.3502 | 5.2867 | 2994.499 | 2994.613 | 40000 |
| Hybrid EM | 3.5 | 0.7 | 17 | 7.3677 | 7.7318 | 3.3513 | 5.2869 | 2995.804 | n.a. | 51989† |
| MAL-DE | 3.5 | 0.7 | 17 | 7.3 | 7.7153 | 3.3502 | 5.2867 | 2994.471 | 2994.471 | 120000 |
| mCDE | 3.5 | 0.7 | 17 | 7.3 | 7.7153 | 3.3502 | 5.2867 | 2994.342 | n.a. | *500*‡ |

† number of function evaluations of the best run; ‡ number of iterations; n.a. means not available.

**Speed Reducer Design Problem**

The weight of the speed reducer is to be minimized subject to the constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stress in the shafts as described in Tomassetti (2010). There are seven variables and 11 inequality constraints. This is a mixed variables problem, where the variable $x_3$ is integer (number of teeth) and the others are continuous. The mathematical formulation of the optimization problem is as follows:

$$\begin{aligned}
&\min\ 0.7854x_1x_2^2((10/3)x_3^2+14.9334x_3-43.0934)-1.508x_1(x_6^2+x_7^2)\\
&\quad +7.4777(x_6^3+x_7^3)+0.7854(x_4x_6^2+x_5x_7^2)\\
&\text{subject to } \frac{27}{x_1x_2^2x_3}-1\le 0,\ \frac{397.5}{x_1x_2^2x_3^2}-1\le 0,\ \frac{1.93x_4^3}{x_2x_3x_6^4}-1\le 0,\\
&\quad \frac{1.93x_5^3}{x_2x_3x_7^4}-1\le 0,\ \frac{x_2x_3}{40}-1\le 0,\ \frac{5x_2}{x_1}-1\le 0,\\
&\quad \frac{x_1}{12x_2}-1\le 0,\ \frac{1.5x_6+1.9}{x_4}-1\le 0,\ \frac{1.1x_7+1.9}{x_5}-1\le 0\\
&\quad \frac{\left((\frac{745x_4}{x_2x_3})^2+16.9\times 10^6\right)^{1/2}}{110.0x_6^3}-1\le 0,\ \frac{\left((\frac{745x_5}{x_2x_3})^2+157.5\times 10^6\right)^{1/2}}{85.0x_7^3}-1\le 0,
\end{aligned}$$

where $2.6\le x_1\le 3.6, 0.7\le x_2\le 0.8, 17\le x_3\le 28, 7.3\le x_4\le 8.3, 7.3\le x_5\le 8.3, 2.9\le x_6\le 3.9$ and $5.0\le x_7\le 5.5$. The comparative results among Shifted-HAL, APS-GA, CPSO, HEA-ACT, Hybrid EM, MAL-DE and mCDE are shown in Table 5. It can be concluded that the proposed algorithm performs reasonably well.

**Coil Compression Spring Design Problem**

This is a real world optimization problem involving discrete, integer and continuous design variables. The objective is to minimize the volume of a spring steel wire used to manufacture the spring (with minimum weight) (see Lampinen and Zelinka (1999)). The design problem has three variables and eight inequality constraints, where $x_1$, the number of coils, is integer, $x_2$, the outside diameter of the spring, is continuous, and $x_3$, the spring wire diameter, is taken from a set of discrete values (see Table 6). The mathematical formulation is as follows:

$$\begin{aligned}
&\min\ \pi^2(x_1+2)x_2x_3^2/4\\
&\text{subject to } \frac{8C_fF_{\max}x_2}{\pi x_3^3}-S\le 0,\ C_f\le 0,\ l_f-l_{\max}\le 0,\\
&\quad \sigma_p-\sigma_{pm}\le 0,\ \sigma_w-\frac{F_{max}-F_p}{K}\le 0,\\
&\quad \sigma_p+\frac{F_{max}-F_p}{K}+1.05(x_1+2)x_3-l_f\le 0,
\end{aligned}$$

where $1\le x_1\le l_{\max}/d_{\min}$, $3d_{\min}\le x_2\le D_{\max}$, $d_{\min}\le x_3\le D_{\max}/3$ and $F_{\max}=1000$ lb., $S=189000$ psi., $l_{\max}=14$ in., $d_{\min}=0.2$ in., $D_{\max}=3.0$ in., $F_p=300$ lb., $G=11.5\times 10^6$, $\sigma_{pm}=6.0$ in., $\sigma_w=1.25$ in. and

$$\begin{aligned}
C_f&=\frac{4(x_2/x_3)-1}{4(x_2/x_3)-4}+\frac{0.615x_3}{x_2},\ K=\frac{Gx_3^4}{8x_1x_2^3},\\
l_f&=\frac{F_{\max}}{K}+1.05(x_1+2)x_3,\ \sigma_p=\frac{F_p}{K}.
\end{aligned}$$

Tests were done using the proposed Shifted-HAL with enhanced AF-2S, and a comparison with the differential evolution method coupled with a soft-constraint approach (DE-sca) (Lampinen and Zelinka 1999), with the mCDE (Azad and Fernandes 2013) and with the ranking selection-based particle swarm optimizer (RSPSO) (J. Wang and Yin 2008) is carried out. The results are shown in Table 7. It can be concluded that the results produced by the algorithm are quite competitive after a reduced computational effort. A low average number of function evaluations has been reported since the algorithm converged after six outer iterations in 20 out of the 30 runs.

Table 6. Available spring steel wire diameters

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.009 | 0.0095 | 0.0104 | 0.0118 | 0.0128 | 0.0132 | 0.014 | 0.015 | 0.0162 | 0.0173 | 0.018 | 0.020 | 0.023 | 0.025 |
| 0.028 | 0.032 | 0.035 | 0.041 | 0.047 | 0.054 | 0.063 | 0.072 | 0.080 | 0.092 | 0.105 | 0.120 | 0.135 | 0.148 |
| 0.162 | 0.177 | 0.192 | 0.207 | 0.225 | 0.244 | 0.263 | 0.283 | 0.307 | 0.331 | 0.362 | 0.394 | 0.4375 | 0.500 |

Table 7. Comparative results for the coil compression spring design problem

| Method | $x_1$ | $x_2$ | $x_3$ | $f_{\text{best}}$ | $f_{\text{av}}$ | n.f.e.$_{\text{av}}$ |
|---|---|---|---|---|---|---|
| Shifted-HAL | 9 | 1.223 | 0.283 | 2.65856 | 2.67220 | 8372 |
| DE-sca | 9 | 1.223 | 0.283 | 2.65856 | n.a. | 26000$^\dagger$ |
| mCDE | 9 | 1.223 | 0.283 | 2.65856 | n.a. | *500*$^\ddagger$ |
| RSPSO | 9 | 1.223 | 0.283 | 2.65856 | 2.72887 | 15000 |

$^\dagger$ number of function evaluations of the best run; $^\ddagger$ number of iterations; n.a. means not available.

**Economic Dispatch with Valve-point Effect Problem**

The economic dispatch (ED) problem reflects an important optimization task in power generation systems. The objective is to find the optimal combination of power dispatches from $n$ different power generating units to minimize the total generation cost while satisfying the specified load demand and the generating units operating conditions. Hence, the objective is

$$\min \sum_{i=1}^{n} a_{i,1}P_i^2 + a_{i,2}P_i + a_{i,3} + \left| a_{i,4} \sin\left(a_{i,5}\left(P_{i\,\min} - P_i\right)\right)\right|, \tag{15}$$

when valve-point loading effects are considered, where $P_i$ is the $i$th unit power output (per hour), $a_{i,1}$, $a_{i,2}$ and $a_{i,3}$ are the cost coefficients of unit $i$, and $a_{i,4}$ and $a_{i,5}$ are the coefficients of unit $i$ reflecting valve-point loading effects (see Hardiansyah (2013); Sinha, Chakrabarti, and Chattopadhyay (2003)). The total power output from $n$ units should be equal to the total load demand, $P_D$, plus the transmission loss, $P_L$, of the system

$$\sum_{i=1}^{n} P_i = P_D + P_L,$$

where $P_L$ is calculated using the power flow coefficients $B_{i,j}$ by:

$$P_L = \sum_{i=1}^{n}\sum_{j=1}^{n} P_i B_{i,j} P_j + \sum_{i=1}^{n} B_{0,i}P_i + B_{0,0}.$$

The power output from unit $i$ should satisfy $P_{i\,\min} \leq P_i \leq P_{i\,\max}$, $i = 1,2,\ldots,n$, where $P_{i\,\min}$ and $P_{i\,\max}$ are the minimum and the maximum real power outputs of the $i$th unit, respectively.

In practice, ramp rate limits restrict the operating range of a unit for adjusting the generator operation between two operating periods. The generation may increase or decrease with corresponding upper and downward ramp rate limits. If the power generation increases then $P_i - P_{i,0} \leq U_i$, otherwise $P_{i,0} - P_i \leq D_i$, where $P_{i,0}$ is the power generation of $i$th unit at the previous hour and $U_i$ and $D_i$ are the upper and lower ramp rate limits, respectively. Due to physical limitations of machine components, the generating units may have certain zones where operation is prohibited. Hence, for the $i$th generating unit, the following condition is required $P_i \leq P_i^{Lpz}$ and $P_i \geq P_i^{Upz}$ where $P_i^{Lpz}$ and $P_i^{Upz}$ are the lower and upper limits of a given prohibited zone for the $i$th unit.

To demonstrate the performance and applicability of the proposed method an instance with

40 generating units (ED-40), a hourly demand of 10500 MW, without power loss, ramp rate limits and prohibited operating zones, is considered. The input data concerned with $a_{i,k}$ ($k = 1,\dots,5$), $P_{i\min}$, $P_{i\max}$, for $i = 1,\dots,40$ are available in Hardiansyah (2013). The Shifted-HAL algorithm is compared with the hybrid genetic algorithm (HGA) based on differential evolution and a sequential quadratic programming (SQP) local search, presented in He, Wang, and Mao (2008), an evolutionary programming technique (EPT) addressed in Sinha, Chakrabarti, and Chattopadhyay (2003), a differential evolution approach combined with chaotic sequences and a SQP gradient-based local search (DEC(2)-SQP(1)) (Coelho and Mariani 2006), the Gaussian and Cauchy distribution-based PSO variant (G-C-PSO) in Hardiansyah (2013), and a modified particle swarm optimization (MPSO) algorithm presented in Park, Lee, Shin, and Lee (2005). The best run (among 30 runs) produced a cost value of 121686.77 after 2500 iterations, 600296 function evaluations and 11.04 seconds – with $k_{\max} = 50$, $t_{\max} = 50$ and $NF_{\max} = 10m$ – (see other results in Table 8) and the optimal dispatch results for the 40 generators are reported in Table 9. It should be noticed that when solving larger dimensional problems, the size of the population may become large since both 'master swarm' and 'training swarm' should have at least $n$ points each to be able to provide $n$ points for the N-M local search. It can be concluded that the Shifted-HAL algorithm produces acceptably good results, noting that HGA outperforms the other methods in comparison since it uses a gradient-based SQP local optimizer.

Table 8. Cost comparative results for the ED-40 problem (in \$/hour)

| Method | $f_{\text{best}}$ | $f_{\text{med}}$ | $f_{\text{av}}$ | St.d.($f$) | n.ite. |
|---|---|---|---|---|---|
| Shifted-HAL$^{\dagger}$ | 121686.77 | 122895.45 | 122966.59 | 5.2E+02 | 2500 |
| DEC(2)-SQP(1)$^{\flat}$ | 121741.98 | n.a. | 122295.13 | 3.9E+02 | 600 |
| EPT | 122624.35 | n.a. | 123382.00 | n.a. | n.a. |
| G-C-PSO | 121649.20 | n.a. | n.a | n.a. | n.a. |
| HGA | 121418.27 | n.a. | 121784.04 | n.a. | 3000$^{\ddagger}$ |
| MPSO$^{\S}$ | 122252.27 | n.a. | n.a | n.a. | n.a. |

$^{\dagger}$ The results were obtained with a population of $m = 200$ points and $\upsilon = 10^{-10}$; $^{\flat}$ population size is set to 30; $^{\ddagger}$ A maximum of 50 DE operator iterations are done at each of the 60 hybrid GA iterations, for a population of 82 points; $^{\S}$ 53% of the runs converged to a value in the range $[122500, 123000]$, the remaining converged to a value in $[120000, 122500]$; n.a. means not available.

Table 9. Dispatch results for the 40 generators (in MW) for comparison

| | $P_1$ —- $P_{10}/P_{11}$ —- $P_{20}/P_{21}$ —- $P_{30}/P_{31}$ —- $P_{40}$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Shifted-HAL | 113.981 | 110.831 | 101.020 | 179.627 | 97.000 | 139.998 | 299.938 | 284.599 | 293.465 | 130.094 |
| | 169.128 | 94.116 | 305.022 | 394.299 | 304.518 | 304.536 | 489.305 | 490.093 | 511.868 | 511.266 |
| | 523.278 | 523.333 | 523.267 | 525.583 | 523.287 | 523.251 | 10.030 | 10.175 | 10.028 | 88.957 |
| | 190.000 | 184.549 | 190.000 | 178.055 | 164.888 | 165.219 | 109.973 | 109.942 | 109.986 | 511.494 |
| G-C-PSO | 113.997 | 112.652 | 119.426 | 189.000 | 96.871 | 139.280 | 223.592 | 284.580 | 216.433 | 239.336 |
| | 314.873 | 305.057 | 365.543 | 493.373 | 280.433 | 432.072 | 435.243 | 417.696 | 532.188 | 409.205 |
| | 534.063 | 457.096 | 441.363 | 397.362 | 446.418 | 442.116 | 74.862 | 27.543 | 76.831 | 97.000 |
| | 118.378 | 188.752 | 190.000 | 120.703 | 170.240 | 198.990 | 110.000 | 109.341 | 109.924 | 468.169 |
| HGA | 111.379 | 110.928 | 97.410 | 179.733 | 89.219 | 140.000 | 259.620 | 284.657 | 284.659 | 130.000 |
| | 168.821 | 168.850 | 214.752 | 394.285 | 304.536 | 394.299 | 489.288 | 489.287 | 511.275 | 511.286 |
| | 523.296 | 523.320 | 523.291 | 523.301 | 523.268 | 523.279 | 10.000 | 10.000 | 10.000 | 88.638 |
| | 190.000 | 190.000 | 190.000 | 164.980 | 165.997 | 165.046 | 110.000 | 110.000 | 110.000 | 511.301 |
| MPSO | 114.000 | 114.000 | 120.000 | 182.222 | 97.000 | 140.000 | 300.000 | 299.021 | 300.000 | 130.000 |
| | 94.000 | 94.000 | 125.000 | 304.485 | 394.607 | 305.323 | 490.272 | 500.000 | 511.404 | 512.174 |
| | 550.000 | 523.655 | 534.661 | 550.000 | 525.057 | 549.155 | 10.000 | 10.000 | 10.000 | 97.000 |
| | 190.000 | 190.000 | 190.000 | 200.000 | 200.000 | 200.000 | 110.000 | 110.000 | 110.000 | 512.964 |

The other instance has 15 generating units (ED-15) and a hourly demand of 2630 MW. The objective function is smooth and the input data concerned with $a_{i,k}$ ($k = 1,\dots,3$), $B_{i,j}$ ($j = 1,\dots,15$), $B_{0,i}$, $B_{0,0}$, $P_{i\min}$, $P_{i\max}$, $U_i$, $D_i$, for $i = 1,\dots,15$, $P_i^{Lpz}$ and $P_i^{Upz}$ (for three zones embedded in units 2, 5, 6 and two zones in unit 12) are available in Coelho and Lee (2008). The results produced by the proposed algorithm are compared with those of a PSO version which is based on Gaussian probability distribution linked with chaotic sequences (ch-G-PSO) in (Coelho and Lee 2008), of a firefly algorithm (FA) in (Yang, Hosseini, and Gandomi 2012), of the improved coordinated aggregation-based PSO (ICA-PSO) by (Vlachogiannis and Lee 2009), of the hybrid bacterial foraging optimization with Nelder-Mead search (BFO-NM) presented in Panigrahi and Pandi (2008), and of a PSO version with normalized evaluation values (norm-PSO) (see Gaing (2003)). The best run (out of 30 runs) of the proposed algorithm produced a cost value of 32692.64 after a total of 300 iterations, 15138 function evaluations and 0.16 seconds (with $k_{\max} = 10$, $t_{\max} = 30$ and $NF_{\max} = 100$). To obtain high quality solutions $\upsilon = 10^{-10}$ and $\delta_i^{(1)} = 0$ for all $i = 1,\ldots,p$ are used. Other results of the experiment are shown in Table 10. The optimal dispatch results for the 15 generators are reported in Table 11. It can be concluded that the Shifted-HAL algorithm is capable of competing with other well-known metaheuristics.

Table 10. Cost comparative results for the ED-15 problem (in \$/hour)

| Method | $f_{\text{best}}$ | $f_{\text{med}}$ | $f_{\text{av}}$ | St.d.($f$) | loss | n.ite. |
|---|---|---|---|---|---|---|
| Shifted-HAL$^\dagger$ | 32692.64 | 32721.78 | 32739.94 | 5.3E+01 | 29.56 | 300 |
| BFO-NM | 32784.50 | n.a. | 32796.8 | 8.5E+01 | n.a. | *10000*$^\S$ |
| ch-G-PSO$^\ddagger$ | 32508.12 | n.a. | 35122.79 | 1.9E+03 | 13.67 | 100 |
| FA | 32704.45 | n.a. | 32856.1 | 1.5E+02 | 30.66 | *50000*$^\S$ |
| ICA-PSO$^\flat$ | 32393.23 | n.a. | 32400.17 | n.a. | 11.45 | 345 |
| norm-PSO$^\natural$ | 32858 | n.a. | 33039 | n.a. | 32.43 | 200 |

$^\dagger$ The results were obtained with a population of $m = 50$ points; $^\S$ number of function evaluations; $^\ddagger$ population size is set to 50; $^\flat$ initial population size is set to 40; $^\natural$ population size is set to 100; n.a. means not available.

Table 11. Dispatch results for the 15 generators (in MW) for comparison

| | $P_1$ —- $P_8$/$P_9$ —- $P_{15}$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Shifted-HAL | 455.0000 | 380.0000 | 130.0000 | 130.0000 | 170.0000 | 460.0000 | 430.0000 | 60.0000 |
| | 69.5554 | 160.0000 | 80.0000 | 80.0000 | 25.0000 | 15.0000 | 15.0000 | |
| ch-G-PSO | 440.4990 | 179.5947 | 21.0524 | 87.1376 | 360.7675 | 395.8330 | 432.0085 | 168.9198 |
| | 162.0000 | 138.4343 | 52.6294 | 66.8875 | 62.7471 | 47.5574 | 27.6065 | |
| FA | 455.0000 | 380.0000 | 130.0000 | 130.0000 | 170.0000 | 460.000 | 430.0000 | 71.7450 |
| | 58.9164 | 160.0000 | 80.0000 | 80.0000 | 25.0000 | 15.0000 | 15.0000 | |

## 5. Conclusions

An enhanced AF-2S algorithm that is implemented within an augmented Lagrangian framework to solve the bound constrained subproblems is presented. A new shifted hyperbolic penalty function with interesting properties is derived and it is proved that a sequence of iterates produced by the shifted hyperbolic augmented Lagrangian algorithm converges to an $\varepsilon$-global minimizer. The enhanced AF-2S algorithm is based on an intensification phase that is invoked with a dynamically defined probability and aimed to explore the search space and to avoid local optimal solution. The reported numerical results show the good performance of the shifted-HAL algorithm. Further testing will be carried out in the future with large-dimensional problems in the context of engineering applications. Furthermore, a deterministic and exact solver for bound constrained global optimization problems, like the Multilevel Coordinate Search of Huyer and Neumaier (1999), will be considered as future developments to be integrated into the Shifted-HAL algorithm.

## Disclosure statement

No potential conflict of interest was reported by the authors.


## Acknowledgements

This work was supported by COMPETE: POCI-01-0145-FEDER-007043, FCT - Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2013 and partially supported by CMAT - Centre of Mathematics of the University of Minho.


## References


Akay, B., and D. Karaboga. 2012.“Artificial bee colony algorithm for large-scale problems and engineering design optimization.” *Journal of Intelligent Manufacturing* 23 (4): 1001–1014.

Ali, M. M., M. Golalikhani, and J. Zhuang. 2014. “A computational study on different penalty approaches for solving constrained global optimization problems with the electromagnetism-like method.” *Optimization* 63 (3): 403–419.

Ali, M. M., and W. X. Shu. 2013. “A penalty function-based differential evolution algorithm for constrained global optimization.” *Computational Optimization and Applications* 54(3), 707–739.

Azad, M. A. K., and E. M. G. P. Fernandes. 2013. “Modified constrained differential evolution for solving nonlinear global optimization problems.” In *Computational Intelligence* (Series: Studies in Computational Intelligence), edited by K. Madani, A. Dourado, A. Rosa, J. Filipe, Vol. 465, 85–100 Springer Berlin Heidelberg.

Azad, M. A. K., A. M. A. C. Rocha, and E. M. G. P. Fernandes. 2014. “Improved binary artificial fish swarm algorithm for the 0–1 multidimensional knapsack problems.” *Swarm and Evolutionary Computation* 14: 66–75.

Barbosa, H. J. C., and A. C. C. Lemonge. 2008. “An adaptive penalty method for genetic algorithms in constrained optimization problems.” In *Frontiers in Evolutionary Robotics*, (ISBN: 978-3-902613-19-6) edited by H. Iba, 9–34. I-Tech Education Publ., Austria.

Bertsekas, D. P. 1999. *Nonlinear Programming*, 2nd edn. Athena Scientific, Belmont.

Birgin, E. G., C. A. Floudas, and J. M. Martínez. 2010. “Global minimization using an Augmented Lagrangian method with variable lower-level constraints.” *Mathematical Programming, Ser. A* 125 (1): 139–162.

Boussaïd, I., J. Lepagnot, and P. Siarry. 2013. “A survey on optimization metaheuristics.” *Information Sciences* 237: 82–117.

Coelho L. S., C.-S. Lee. 2008. “Solving economic load dispatch problems in power systems using chaotic and Gaussian particle swarm optimization approaches.” *Electrical Power and Energy Systems* 30 : 297–307.

Coelho L. S., V. C. Mariani. 2006. “Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect.” *IEEE Transactions on Power Systems* 21 (2): 989–996.

Coello, C. A. C. 2002. “Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art.” *Computer Methods in Applied Mechanics and Engineering* 191 (11–12): 1245–1287.

Costa, L., I. A. C. P. Espírito Santo, and E. M. G. P. Fernandes. 2012. “A hybrid genetic pattern search augmented Lagrangian method for constrained global optimization.” *Applied Mathematics and Computation* 218 (18): 9415–9426.

Costa, M. F. P., A. M. A. C. Rocha, and E. M. G. P. Fernandes. 2014. “An artificial fish swarm algorithm based hyperbolic augmented Lagrangian method.” *Journal of Computational and Applied Mathematics* 259 (Part B): 868–876.

Deb, K., and S. Srivastava. 2012. “A genetic algorithm based augmented Lagrangian method for constrained optimization.” *Computational Optimization and Applications* 53 (3): 869–902.

Di Pillo, G., S. Lucidi, and F. Rinaldi. 2012. "An approach to constrained global optimization based on exact penalty functions." *Journal of Global Optimization* 54 (2): 251–260.

Gaing, Z.-L. 2003. "Particle swarm optimization to solving the economic dispatch considering the generator constraints." *IEEE Transactions on Power Systems* 18 (3): 1187–1195.

Gonzaga, C. C., and R. A. Castillo. 2003. "A nonlinear programming algorithm based on non-coercive penalty functions." *Mathematical Programming, Ser. A* 96 (1): 87–101.

Hardiansyah. 2013. "A modified particle swarm optimization technique for economic load dispatch with valve-point effect." *International Journal of Intelligent Systems and Applications* 7: 32–41.

He, D., F. Wang, and Z. Mao. 2008. "A hybrid genetic algorithm approach based on differential evolution for economic dispatch with valve-poin effect." *Electrical Power and Energy Systems* 30: 31–38.

Hedar, A.-R., and M. Fukushima. 2006. "Derivative-free filter simulated annealing method for constrained continuous global optimization." *Journal of Global Optimization* 35 (4): 521–549.

Huyer, W., and A. Neumaier. 1999. "Global optimization by multilevel coordinate search." *Journal of Global Optimization* 14: 331–355.

Jansen, P. W., and R. E. Perez. 2011. "Constrained structural design optimization via a parallel augmented Lagrangian particle swarm optimization approach." *Computers & Structures* 89 (13-14): 1352–1366.

Kamarian, S., M. H. Yasb, A. Pourasgharc, and M. Daghagh. 2014. "Application of firefly algorithm and ANFIS for optimisation of functionally graded beams." *Journal of Experimental & Theoretical Artificial Intelligence* 26 (2): 197–209.

Kennedy, J., and R. Eberhart. 2001. *Swarm Intelligence*. San Mateo, CA: Morgan Kaufmann.

Lampinen, J., and I. Zelinka. 1999. "Mixed integer-discrete-continuous optimization by differential evolution - Part 2: a practical example." In *Proceedings of the 5th International Mendel Conference on Soft Computing*, 77–81, Faculty of Mechanical Engineering, Institute of Automation and Computer Science, Brno (Czech Republic).

Lee, C., and X. Yao. 2004. "Evolutionary programming using the mutations based on the Lévy probability distribution." *IEEE Transaction on Evolutionary Computation* 8 (1): 1–13.

Lee, K. S., and Z. W. Geem. 2005. "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice." *Computer Methods in Applied Mechanics and Engineering* 194: 3902–3933.

Lemonge, A. C. C., H. J. C. Barbosa, and H. S. Bernardino. 2015. "Variants of an adaptive penalty scheme for steady-state genetic algorithms in engineering optimization." *Engineering Computations: International Journal for Computer-Aided Engineering and Software* 32 (8): 2182–2215.

Lobato, F. S., and V. Steffen Jr. 2014. "Fish swarm optimization algorithm applied to engineering system design." *Latin American Journal of Solids and Structures* 11 (1): 143–156.

Long, W., X. Liang, Y. Huang, and Y. Chen. 2013. "A hybrid differential evolution augmented Lagrangian method for constrained numerical and engineering optimization." *Computer-Aided Design* 45 (12): 1562–1574.

Mahdavi, M., M. Fesanghary, and E. Damangir. 2007. "An improved harmony search algorithm for solving optimization problems." *Applied Mathematics and Computation* 188 (2): 1567–1579.

Mallipeddi, R., and P. N. Suganthan. 2010. "Problem Definitions and Evaluation Criteria for the CEC 2010 Competition on Constrained Real-Parameter Optimization." Technical Report, Nanyang Technological University, Singapore.

Mallipeddi, R., and P. N. Suganthan. 2010. "Differential evolution with ensemble of constraint handling techniques for Solving CEC 2010 benchmark problems." In *Proceedings of the IEEE Congress on Evolutionary Computation* (CEC 2010), Barcelona, IEEE Press, pp. 1907-1914.

McKinnon, K. I. M. 1998. "Convergence of the Nelder–Mead simplex method to a nonstationary point." *SIAM Journal on Optimization* 9 (1): 148–158.

Mezura-Montes, E., and C. A. C. Coello. 2011. "Constraint-handling in nature-inspired numerical optimization: Past, present and future." *Swarm and Evolutionary Computation* 1 (4): 173–194.

Nelder, J. A., and R. Mead. 1965. "A simplex method for function minimization." *The Computer Journal* 7 (4): 308–313.

Neshat, M., G. Sepidnam, M. Sargolzaei, and A. N. Toosi. 2014. "Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications." *Artificial Intelligence Review* 42 (4): 965–997.

Panigrahi, B. K., and V. R. Pandi. 2008. "Bacterial foraging optimization: Nelder-Mead hybrid algorithm for economic load dispatch." *IET Generation, Transmission & Distribution* 2 (4): 556–565.

Park, J.-B., K.-S. Lee, J.-R. Shin, and K. Y. Lee. 2005. "A particle swarm optimization for economic dispatch with nonsmooth cost function." *IEEE Transactions on Power System* 20 (1): 34–42.

Rocha, A. M. A. C., and E. M. G. P. Fernandes. 2009. "Hybridizing the electromagnetism-like algorithm with descent search for solving engineering design problems." *International Journal of Computer Mathematics* 86 (10–11): 1932–1946.

Rocha, A. M. A. C., and E. M. G. P. Fernandes. 2011. "Numerical study of augmented Lagrangian algorithms for constrained global optimization." *Optimization* 60 (10–11): 1359–1378.

Rocha, A. M. A. C., E. M. G. P. Fernandes, and T .F. M. C. Martins. 2011. "Novel fish swarm heuristics for bound constrained global optimzation problems." In *Computational Science and Its Applications*, ICCSA 2011, edited by B. Murgante et al., Lecture Notes in Computer Science, Vol. 6784, Part III, 185–199 Springer Berlin Heidelberg.

Rocha, A. M. A. C., T .F. M. C. Martins, and E. M. G. P. Fernandes. 2011. "An augmented Lagrangian fish swarm based method for global optimization." *Journal of Computational and Applied Mathematics* 235 (16): 4611–4620.

Silva, E. K., H. J. C. Barbosa, and A. C. C. Lemonge. 2011. "An adaptive constraint handling technique for differential evolution with dynamic use of variants in engineering optimization." *Optimization and Engineering* 12 (1–2): 31–54.

Sinha, N., R. Chakrabarti, and P. K. Chattopadhyay. 2003. "Evolutionary programming techniques for economic load dispatch. " *IEEE Transactions on Evolutionary Computation* 7 (1): 83–94.

Tahk, M.-J., H.-W. Woo, and M.-S. Park. 2007. "A hybrid optimization method of evolutionary and gradient search." *Engineering Optimization* 39 (1): 87–104.

Takahama, T., and S. Sakai. 2010. "Constrained optimization by the epsilon constrained differential evolution with an archive and gradient-based mutation." In *Proceedings of IEEE Congress on Evolutionary Computation* (CEC 2010), Barcelona, IEEE Press, pp.1680–1688.

Tomassetti, G. 2010. "A cost-effective algorithm for the solution of engineering problems with particle swarm optimization." *Engineering Optimization* 42 (5): 471–495.

Tsai, H.-C., and Y.-H. Lin. 2011. "Modification of the fish swarm algorithm with particle swarm optimization formulation and communication behavior." *Applied Soft Computing* 11 (8): 5367–5374.

Tsoulos, I.G. 2009. "Solving constrained optimization problems using a novel genetic algorithm." *Applied Mathematics and Computation* 208 (1): 273–283.

Vlachogiannis, J. G., and K. Y. Lee. 2009. "Economic load dispatch – A comparative study on heuristic optimization techniques with an improved coordinated aggregation–based PSO." *IEEE Transactions on Power Systems* 24 (2): 991–1001.

Wang, C.-Y., and D. Li. 2009. "Unified theory of augmented Lagrangian methods for constrained global optimization." *Journal of Global Optimization* 44 (3): 433–458.

Wang, J., and Z. Yin. 2008. "A ranking selection-based particle swarm optimizer for engineering design optimization problems." *Structural and Multidisciplinary Optimization* 37 (2): 131–147.

Wang, Y., Z. Cai, Y. Zhou, and Z. Fan. 2009. "Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique." *Structural and Multidisciplinary Optimization* 37 (4): 395–413.

Xavier, A. E. 2001. "Hyperbolic penalty: a new method for nonlinear programming with inequalities." *International Transactions in Operational Research* 8 (6): 659–671.

Yang, X.-S. , S. S. S. Hosseini, and A. H. Gandomi. 2012. "Firefly algorithm for solving nonconvex economic dispatch problems with valve loading effect." *Applied Soft Computing* 12 (3): 1180–1186.

Yazdani, D., B. Saman, A. Sepas-Moghaddam, F. Mohammad-Kazemi, and M. R. Meybodi. 2013. "A new algorithm based on improved artificial fish swarm algorithm for data clustering." *International Journal of Artificial Intelligence* 11 (A13): 193–221.

Zhang, K., and J. Luo. 2013. "Research on flatness errors evaluation based on artificial fish swarm algorithm and Powell method." *International Journal of Computing Science and Mathematics* 4 (4): 402–411.

Zhao, W., Z. Tang, Y. Yang, L. Wang, and S. Lan. 2014. "Cooperative search and rescue with artificial fishes based on fish-swarm algorithm for underwater wireless sensor networks." *The Scientific World Journal* 2014, Article ID 145306, 10 pages.

Zhou, Y. Y., and X. Q. Yang. 2012. "Augmented Lagrangian functions for constrained optimization problems." *Journal of Global Optimization* 52 (1): 95–108.